[redacted]

1106/1108

TROUBLE SHOOTING GUIDE

(INCORPORATING SYSTEMS 1100/10 + 1100/20)

TZ16000

COMPANY CONFIDENTIAL INFORMATION

THIS DOCUMENT CONTAINS CONFIDENTIAL INFORMATION OF THE SPERRY RAND CORPORATION. IN CONSIDERATION OF THE RECEIPT OF THIS DOCUMENT, THE RECIPIENT AGREES NOT TO REPRODUCE, COPY, USE OR TRANSMIT THIS DOCUMENT AND/OR THE INFORMATION THEREIN CONTAINED, IN WHOLE OR IN PART, OR TO SUFFER SUCH ACTION BY OTHERS, FOR ANY PURPOSE, EXCEPT WITH THE WRITTEN PERMISSION, FIRST OBTAINED, OF SPERRY UNIVAC DIVISION OF SPERRY RAND CORPORATION, AND FURTHER AGREES TO SURRENDER SAME TO SPERRY UNIVAC WHEN THE REASON FOR ITS RECEIPT HAS TERMINATED.

SPERRY UNIVAC
COMPUTER SYSTEMS

PRINTED IN SWITZERLAND

EUROPEAN EDUCATION AND TRAINING CENTER

EUROPEAN EDUCATION AND TRAINING CENTER

ACKNOWLEDGEMENT

This print of the Troubleshooting Guide TZ16000 represents a complete technical revision of the contents and the addition of new material to cover the 1100/10/20. This material in Section 1, together with the revised test procedures in Section 5 will ensure that the Guide has wide and continuing use in the field.

All the revisions and additions were prepared by one man; Gerd Mueller of Swiss Technical Support. EETC wish to extend their thanks to him for his tremendous voluntary effort.

The book was prepared for print by Training Services.

J. CLARKE

EUROPEAN EDUCATION AND TRAINING CENTER

1106 – 1108 – 1100/10 – 1100/20

TROUBLE SHOOTING GUIDE

TZ16000
(TZ20000)

CONTENTS LIST

SECTION 1	INITIAL LOAD TROUBLESHOOTING
SECTION 2	STATIC CHECKS MAIN CONTROL
SECTION 3	ICR
SECTION 4	INPUT/OUTPUT
SECTION 5	TEST PROGRAMS AND LOOPS
SECTION 6	MARGIN VOLTAGE ADJUSTMENTS
SECTION 7	CENTRAL PROCESSOR
SECTION 8	MEMORY

EUROPEAN EDUCATION AND TRAINING CENTER

SECTION 1

INITIAL LOAD TROUBLE SHOOTING

CONTENTS LIST

		Page No.
1.1	PURPOSE	1
1.2	INTRODUCTION	1
1.3	PROGRAM LOADING	1
1.3.1	Tape Format	1
1.3.2	Binary Command/Arithmetic Test	2
1.4	PRELIMINARIES TO INITIAL LOADING	2
	FLOW CHARTS	5–18
1.5	INITIAL LOAD TROUBLE SHOOTING ON 1100/10/20	19
	FLOW CHARTS	20–37

TABLES

1.1	JUMP AND STOP KEYS	3

1 INITIAL LOAD TROUBLE SHOOTING

1.1 PURPOSE

The following information is intended to supply you with a better vehicle of troubleshooting, when faced with an INITIAL LOAD problem.

References in this document are made to "1108/1108A BOOTSTRAP LOADER and SERIAL EXECUTIVE" drawing no. 4096619 Revisions AE, AF and AG.

1.2 INTRODUCTION

This troubleshooting guide uses the aid of the PRELUDE TEST which is contained on the 1st block of the boot tape.

A simplified command test has been incorporated into the 2000_8 word boot block. The processor will first determine its CPU number and its memory bias (MSR) and then it will make one pass through the prelude command test prior to loading any programs, (the prelude test is by-passed when loading the binary command test utilizing jump 12). This test was designed to check that the data, read from a tape subsystem, came in correctly and it also exercises all of the instructions used by the initial loader. If an error is detected, the CPU will come to a zero stop.

It is important to realize that Data Transfer errors (Data read from a Tape Subsystem) may not check because any such error will corrupt the test.

1.3 PROGRAM LOADING

A detailed procedure for loading with an error condition is attached. However, to use these documents a good knowledge of the Initial Load sequence is essential.

Bootstrap Function

The Bootstrap loader utilizes the 1108 bootstrap feature via the magnetic tape subsystem. When a bootstrap function is sent to the tape control, unit Ø will read forward 1 block and the processor stores 2000 words in memory starting at address Ø or some multiple of 32K. When the buffer is full, an Input Monitor is generated. On the 1108, the input monitor causes the processor to start executing instructions beginning at the input monitor location 0220. The 1108A option contains a NOP at this location and control or address is left with the P-register. In the 1108A, P = MSR X $100K_8$.

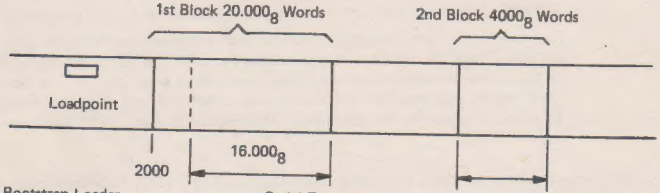
1.3.1 Tape Format

The Bootstrap consists of two blocks on tape (0 – 2000_8 and 3400_8 – 37177_8). The first block contains 2000_8 words of binary information. This block contains a loader for the second block and the C/A test in binary form (2000 to 20000_8). The second block contains additional Bootstrap Loader data (binary form also). The blocks can be loaded with a minimum of processor operation and used for initial debugging.

1.3.2 Binary Command/Arithmetic Test

The Binary C/A Test is written on the boot tape as part of the first block on tape. It is loaded by reloading the input access control word with the correct value while the first block is still being read from the tape as a result of the Bootstrap action. As a result, no I/O instructions or unpack routines are needed to utilize this test. At the end of the buffer, control is given to the C/A Test.

The above statement is not quite true since the I/O channel is closed down when the monitor interrupt occurs. However, it can be reactivated (if JK12) before the next word arrives (~90μsec). To do this reactivation an LIC or LICM-Instruction is required.



Bootstrap Loader
Loaded during
Initial Load.
Includes
Prelude Test

Serial Executive
Binary C/A loaded
by JK12 or JK15
and C/A selected

1108 Loader
Used to load
ICR, CORE, CA Tests
Loaded into Core
$34.000 - 37.777_8$

1.4 PRELIMINARIES TO INITIAL LOADING

- Power Supply Voltages correct
- Power on to all units:
  CPU, Memories, Console
  5008, Uniservo or
  MSA, 5017, Uniservo
- Options selected correctly

CPU:
- Boot Channel (=5008 or MSA)
- Chan. Mode
- D.P. (Double Prec.) not disabled
- LAR selected to last module available
- Boot Ch. not ESI
- Disables

5008:
- not Test Mode
- 7/9 Channel Boot
- Progr. Ø to Boot-Uniservo
- Function received must correspond with servo selected e.g. 7/9 Track etc.

Keys		Boot loader	C/A	Core	ICR	I/O	DVR Exec.
JUMP		Load C/A core ICR & EXEC					
	1	Load via P-Reg.	Delete error				I/O Timing to a prog.
	2	Load EXEC					
	3	Load from servo 1	SP5	SP5	SP5	SP5	SP5
	4	Memory protect	Repeat subtest				
	5		Bypass doub. prec.				
	6		ONE ERROR PER TEST ONLY				
	7		7003 Mem. no Q-Word	1108 no Q-Word	1108	1108	
	8	Create boot tape	Repeat section	REPEAT SUBROUTINE		Select via P-Reg.	
	9		REPEAT PROGRAM			Repeat Channel	
	10	Dump on HSP				Spec. Test	
	11	Dump on 1004 Word adapt.		Reselect Stacks		Spec. word count test	Re-enter C/U/Type
	12		1108, no P17 and no IFR8		TEST addr. 00—77		
	13	Dump on 1004 Char. Adapt.		Error Print Test Addr. on HSP or 1004 100—177			Re-enter inst. timing test
	14	Load name		Delete block X-fer			Cycle on last I/O
	15	Select tests via console					Error Type-Out on HSP
STOP	0	Prelude Test ERROR		1108 A Contr. Stops			
	1	Stop to Modify	STOP AFTER ERROR				Stop before I/O
	2		STOP AFTER TEST				
	3		Stop after Section			Stop after channel	
	4		STOP AFTER CYCLE				

Table 1.1: Jump and Stop Keys

EUROPEAN EDUCATION AND TRAINING CENTER

MSA: – Boot Option patched correctly !

– Device Address - Switches 00–07, Pos Ø

– On-Line, all switches normal

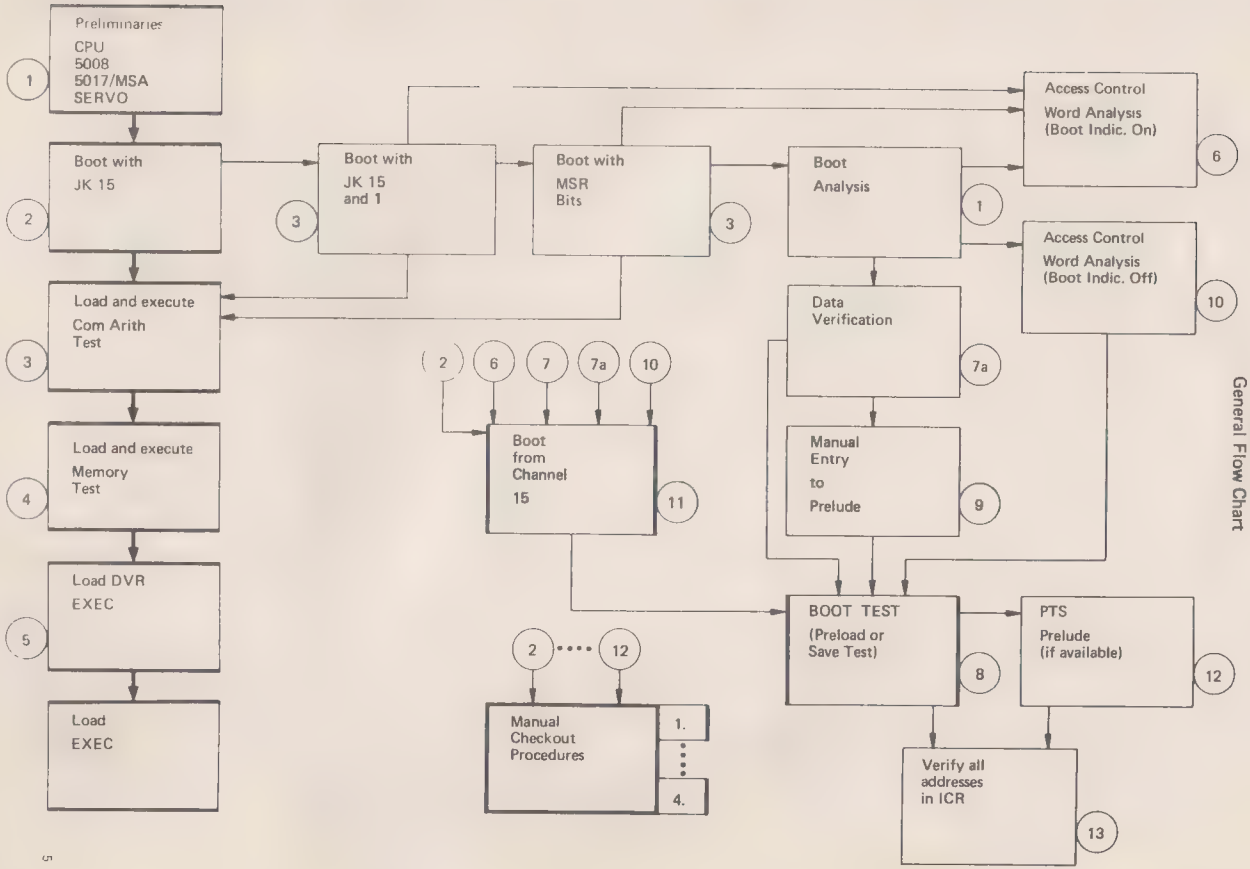
– Function to MSA must correspond with servo selected e.g. 7/9 Track etc.

5017: – Patching (Device Address) must correspond to one selected in MSA

Uniservo:

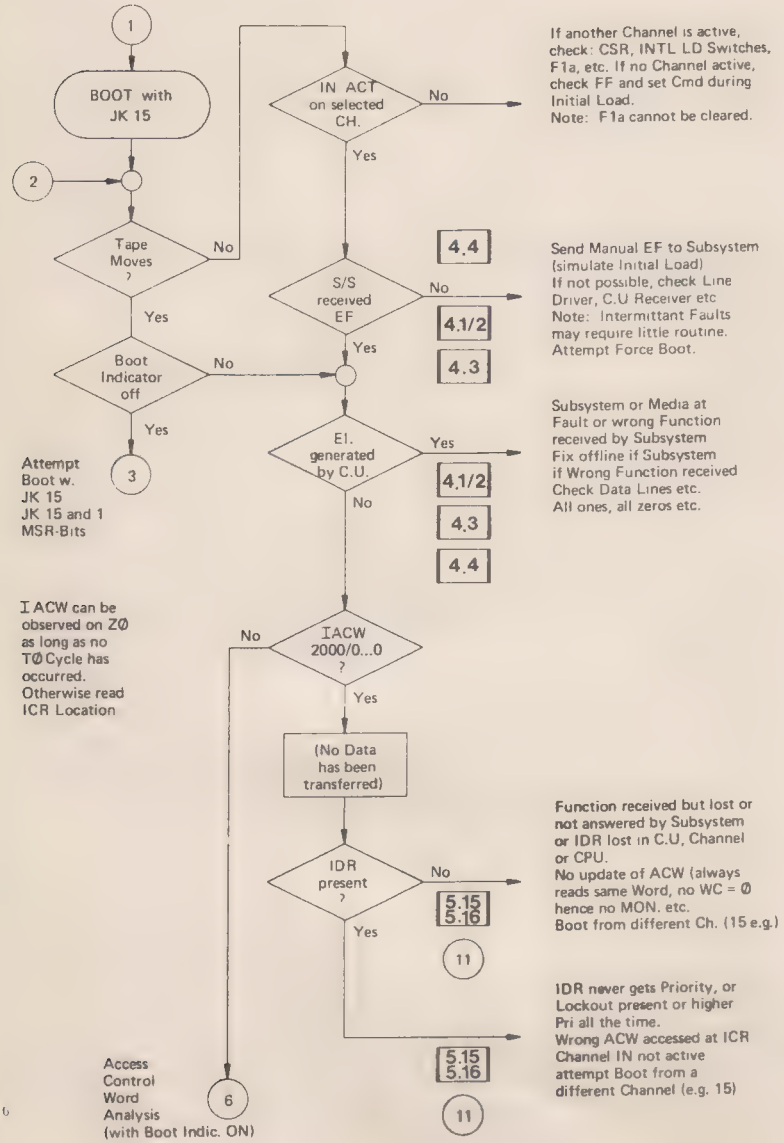
– On, Ready

– Correct Tape (7/9 Ch., Boot Info)

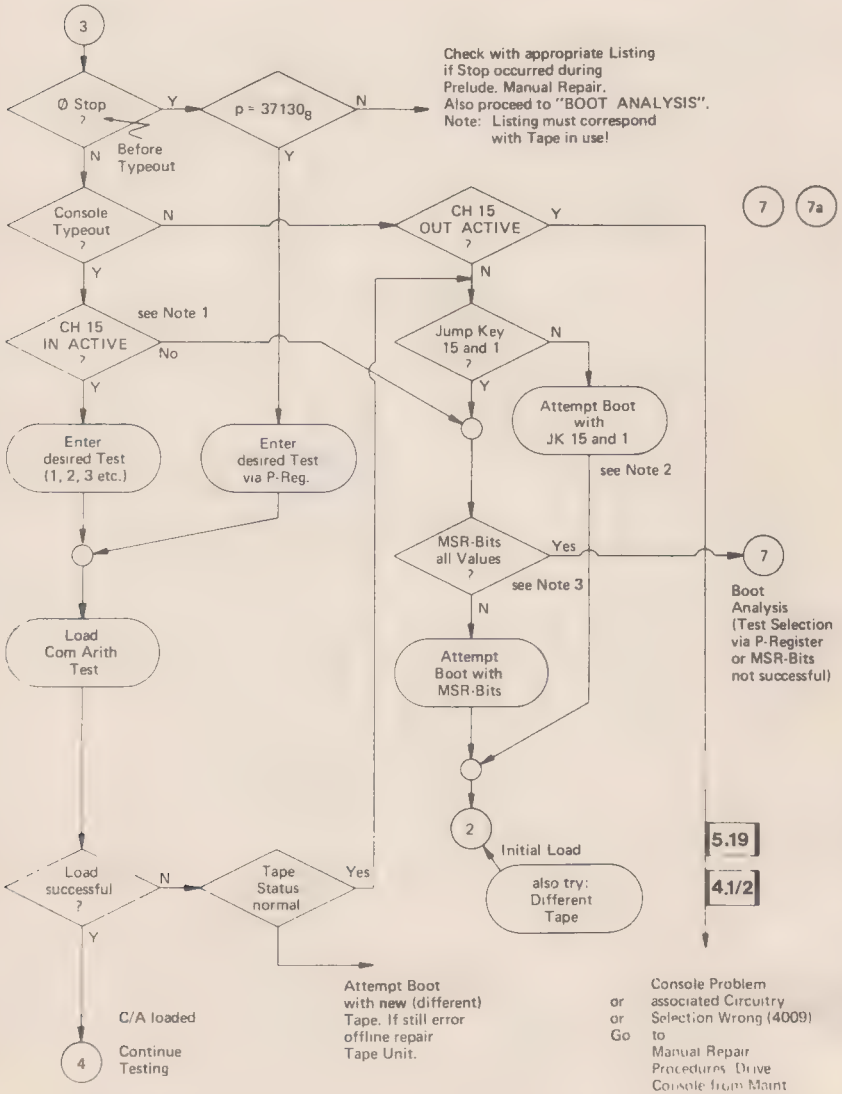


Initial Load
General Flow Chart

Initial Load: TAPE DOES NOT MOVE


1
BOOT with JK 15
2
Tape Moves ?
No
Yes
Boot Indicator off
No
Yes
Attempt Boot w. JK 15 JK 15 and 1 MSR-Bits
3
IN ACT on selected CH.
No
Yes
4.4
S/S received EF
No
4.1/2
Yes
4.3
EI. generated by C.U.
Yes
4.1/2
4.3
4.4
No
IACW 2000/0...0 ?
No
Yes
(No Data has been transferred)
IDR present ?
No
5.15 5.16
11
Yes
5.15 5.16
11
Access Control Word Analysis (with Boot Indic. ON)
6
If another Channel is active, check: CSR, INTL LD Switches, F1a, etc. If no Channel active, check FF and set Cmd during Initial Load.
Note: F1a cannot be cleared.
Send Manual EF to Subsystem (simulate Initial Load)
If not possible, check Line Driver, C.U Receiver etc
Note: Intermittant Faults may require little routine.
Attempt Force Boot.
Subsystem or Media at Fault or wrong Function received by Subsystem
Fix offline if Subsystem if Wrong Function received
Check Data Lines etc.
All ones, all zeros etc.
IACW can be observed on ZØ as long as no TØ Cycle has occurred.
Otherwise read ICR Location
Function received but lost or not answered by Subsystem or IDR lost in C.U, Channel or CPU.
No update of ACW (always reads same Word, no WC = Ø hence no MON. etc.
Boot from different Ch. (15 e.g.)
IDR never gets Priority, or Lockout present or higher Pri all the time.
Wrong ACW accessed at ICR
Channel IN not active
attempt Boot from a different Channel (e.g. 15)

Initial Load: C/A DOES NOT LOAD

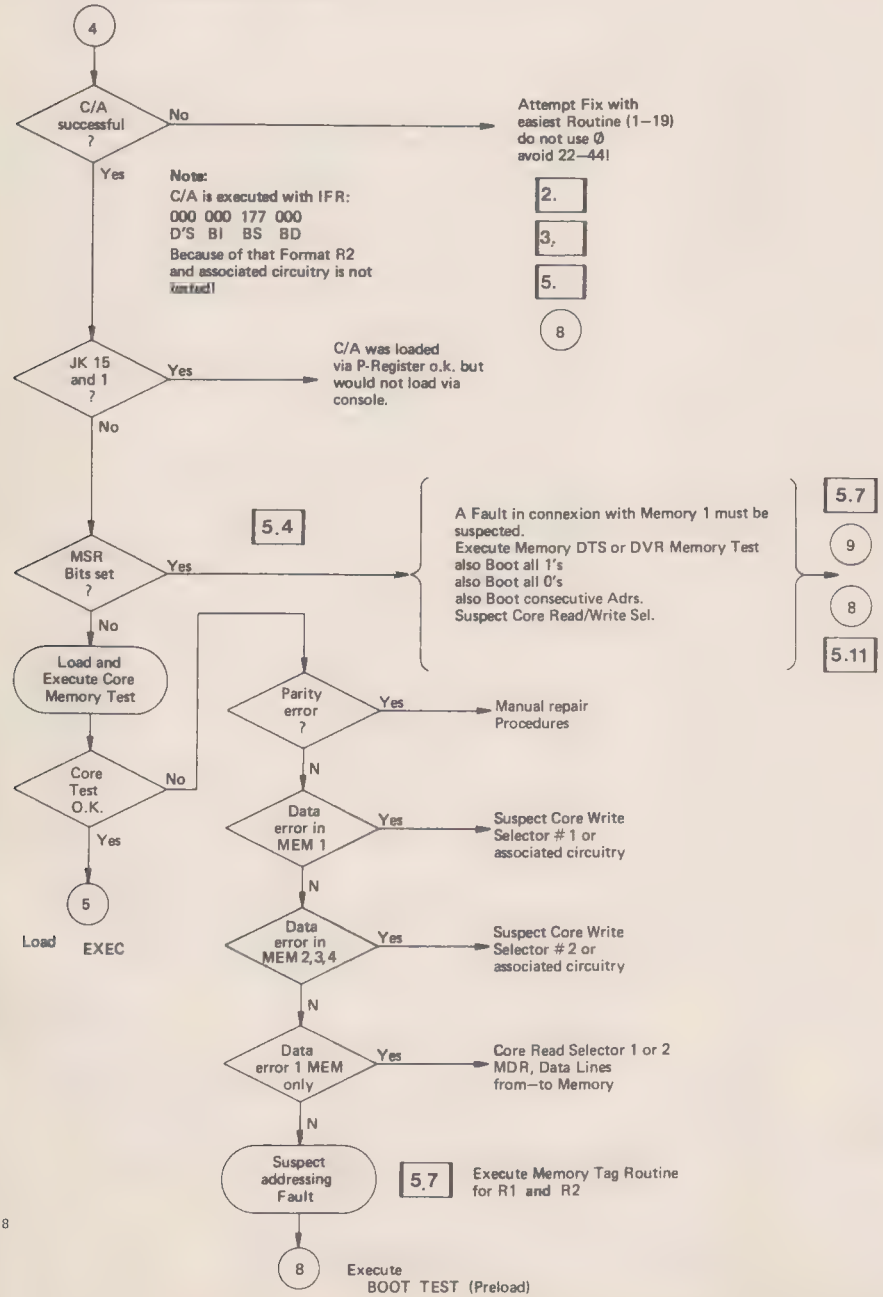


Notes

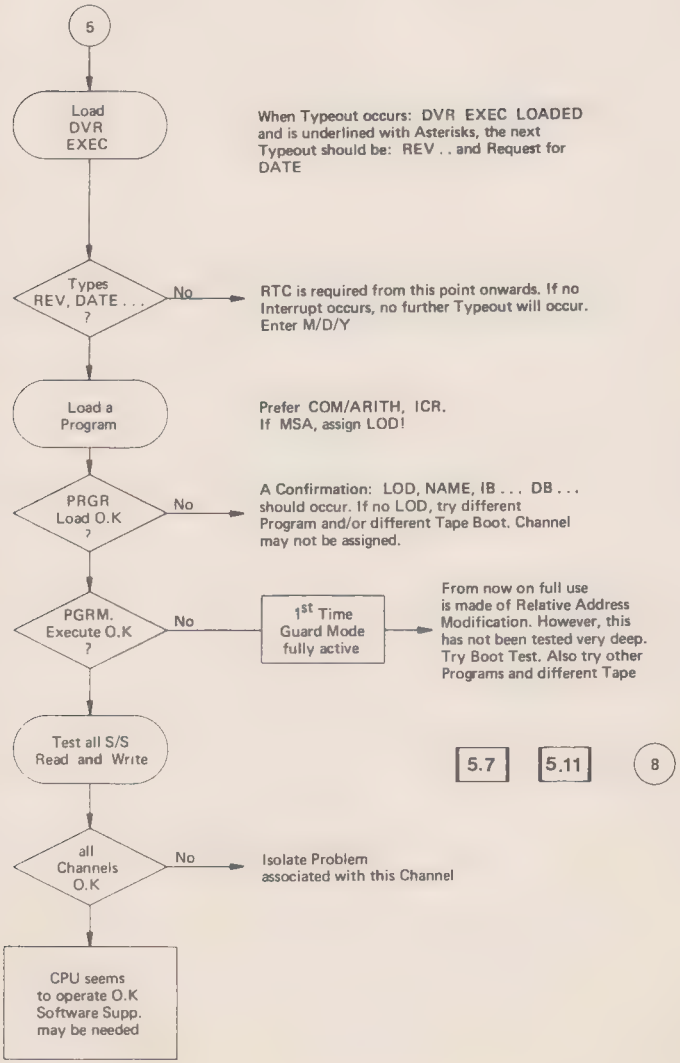
1. Channel may not be able to set IN ACTIVE.
2. All Typeouts are suppressed and Test is selected by entering → P-Reg.
3. DVR AF and AG will not Boot with MSR-Bits unless a Modification of the program is made! Hence go to ⑦ directly

EUROPEAN EDUCATION AND TRAINING CENTER

C/A LOADED, CONTINUE TESTING

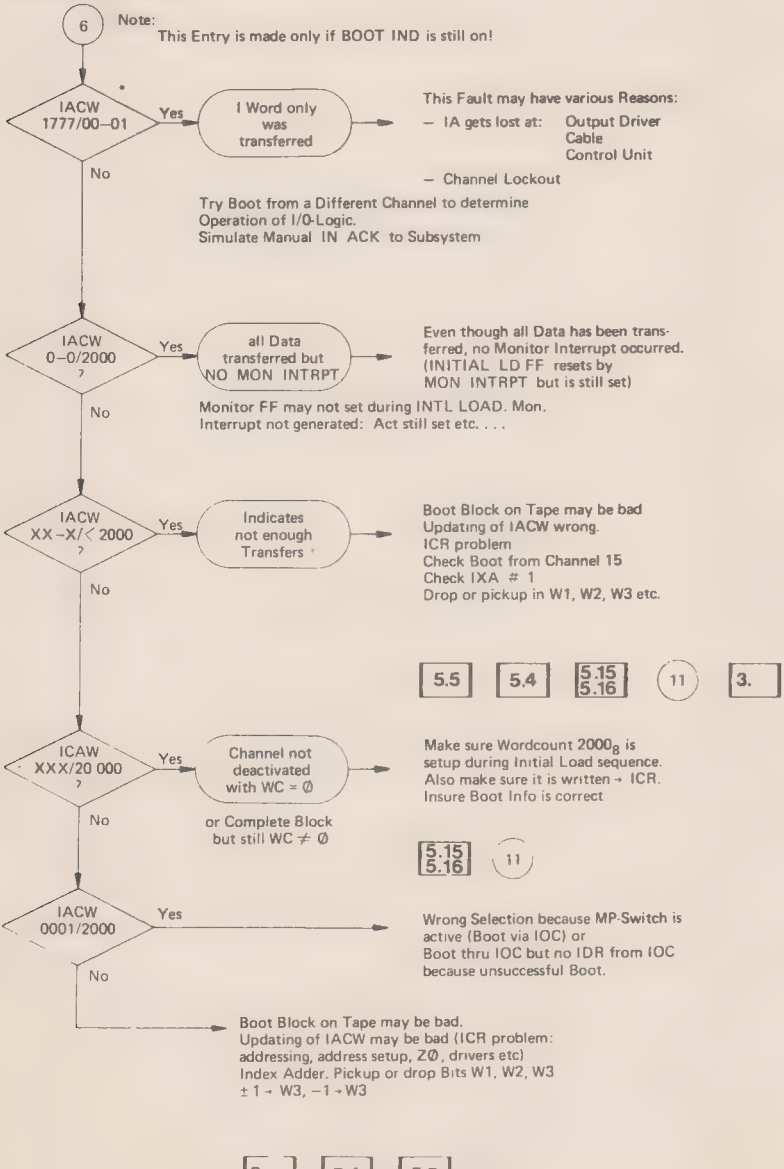


Load DVR EXEC

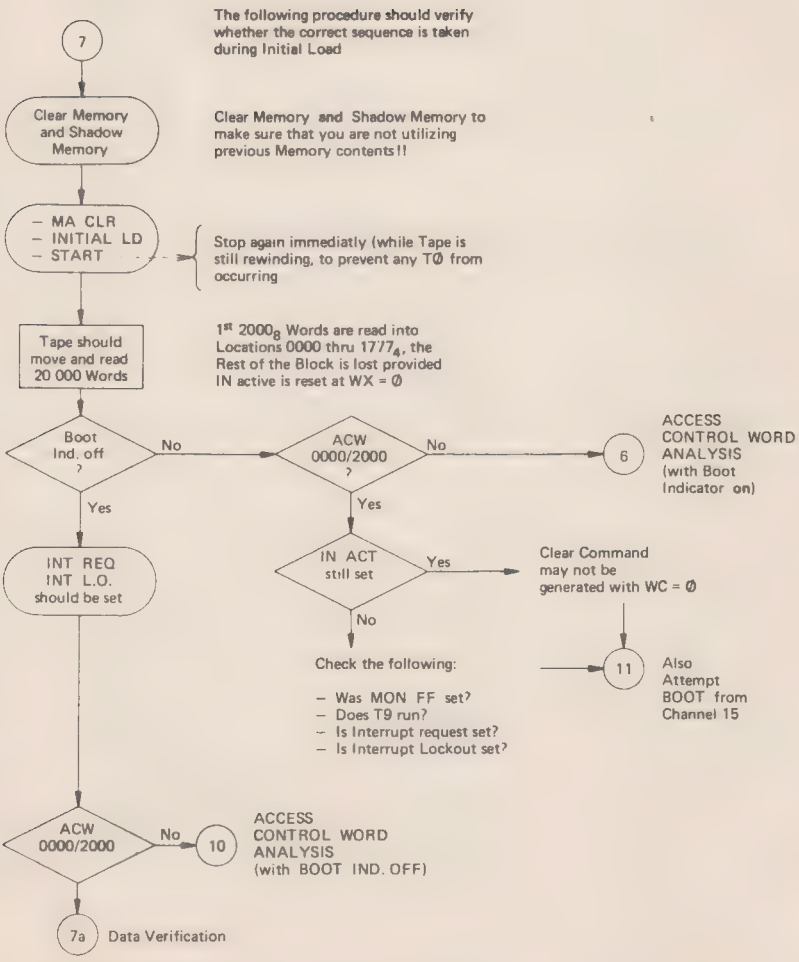

5
Load DVR EXEC
When Typeout occurs: DVR EXEC LOADED and is underlined with Asterisks, the next Typeout should be: REV . . and Request for DATE
Types REV, DATE . . . ?
No
RTC is required from this point onwards. If no Interrupt occurs, no further Typeout will occur. Enter M/D/Y
Load a Program
Prefer COM/ARITH, ICR. If MSA, assign LOD!
PRGR Load O.K ?
No
A Confirmation: LOD, NAME, IB . . . DB . . . should occur. If no LOD, try different Program and/or different Tape Boot. Channel may not be assigned.
PGRM. Execute O.K ?
No
1st Time Guard Mode fully active
From now on full use is made of Relative Address Modification. However, this has not been tested very deep. Try Boot Test. Also try other Programs and different Tape
Test all S/S Read and Write
5.7
5.11
8
all Channels O.K
No
Isolate Problem associated with this Channel
CPU seems to operate O.K Software Supp. may be needed

ACCESS CONTROL WORD ANALYSIS

with BOOT IND ON

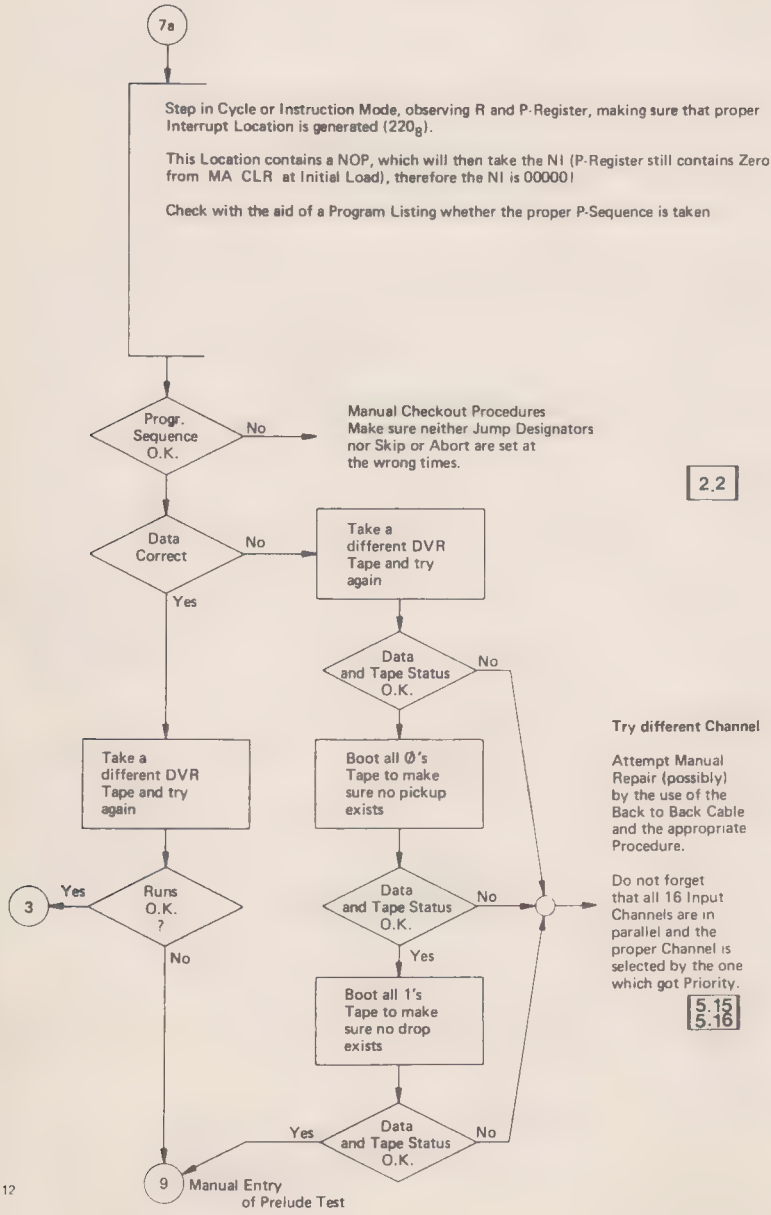

6
Note:
This Entry is made only if BOOT IND is still on!
IACW 1777/00–01
Yes
1 Word only was transferred
This Fault may have various Reasons:
– IA gets lost at: Output Driver Cable Control Unit
– Channel Lockout
No
Try Boot from a Different Channel to determine Operation of I/O-Logic.
Simulate Manual IN ACK to Subsystem
IACW 0–0/2000 ?
Yes
all Data transferred but NO MON INTRPT
Even though all Data has been transferred, no Monitor Interrupt occurred. (INITIAL LD FF resets by MON INTRPT but is still set)
No
Monitor FF may not set during INTL LOAD. Mon. Interrupt not generated: Act still set etc. . . .
IACW XX–X/< 2000 ?
Yes
Indicates not enough Transfers
Boot Block on Tape may be bad
Updating of IACW wrong.
ICR problem
Check Boot from Channel 15
Check IXA # 1
Drop or pickup in W1, W2, W3 etc.
No
5.5
5.4
5.15 5.16
11
3.
ICAW XXX/20 000 ?
Yes
Channel not deactivated with WC = Ø
Make sure Wordcount 2000_8 is setup during Initial Load sequence.
Also make sure it is written → ICR.
Insure Boot Info is correct
No
or Complete Block but still WC ≠ Ø
5.15 5.16
11
IACW 0001/2000
Yes
Wrong Selection because MP-Switch is active (Boot via IOC) or
Boot thru IOC but no IDR from IOC because unsuccessful Boot.
No
Boot Block on Tape may be bad.
Updating of IACW may be bad (ICR problem: addressing, address setup, ZØ, drivers etc)
Index Adder. Pickup or drop Bits W1, W2, W3
± 1 → W3, −1 → W3
3.
5.4
5.5
11

BOOT ANALYSIS

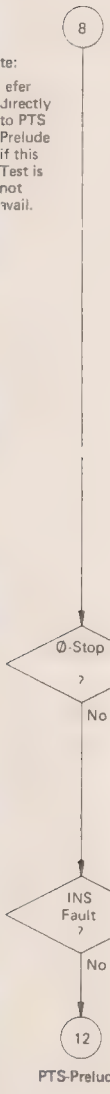

7
The following procedure should verify whether the correct sequence is taken during Initial Load
Clear Memory and Shadow Memory
Clear Memory and Shadow Memory to make sure that you are not utilizing previous Memory contents!!
– MA CLR
– INITIAL LD
– START
Stop again immediatly (while Tape is still rewinding, to prevent any TØ from occurring
Tape should move and read 20 000 Words
1st 2000₈ Words are read into Locations 0000 thru 1777₄, the Rest of the Block is lost provided IN active is reset at WX = Ø
Boot Ind. off ?
No
ACW 0000/2000 ?
No
6
ACCESS CONTROL WORD ANALYSIS (with Boot Indicator on)
Yes
Yes
INT REQ INT L.O. should be set
IN ACT still set
Yes
Clear Command may not be generated with WC = Ø
No
11
Also Attempt BOOT from Channel 15
Check the following:
– Was MON FF set?
– Does T9 run?
– Is Interrupt request set?
– Is Interrupt Lockout set?
ACW 0000/2000
No
10
ACCESS CONTROL WORD ANALYSIS (with BOOT IND. OFF)
7a
Data Verification

EUROPEAN EDUCATION AND TRAINING CENTER

DATA VERIFICATION
(BOOT ALL ONES, ALL ZEROES)


7a
Step in Cycle or Instruction Mode, observing R and P-Register, making sure that proper Interrupt Location is generated (220_8).
This Location contains a NOP, which will then take the NI (P-Register still contains Zero from MA CLR at Initial Load), therefore the NI is 00000!
Check with the aid of a Program Listing whether the proper P-Sequence is taken
Progr. Sequence O.K.
No
Manual Checkout Procedures
Make sure neither Jump Designators nor Skip or Abort are set at the wrong times.
2.2
Data Correct
No
Take a different DVR Tape and try again
Yes
Data and Tape Status O.K.
No
Take a different DVR Tape and try again
Boot all Ø's Tape to make sure no pickup exists
Try different Channel
Attempt Manual Repair (possibly) by the use of the Back to Back Cable and the appropriate Procedure.
Do not forget that all 16 Input Channels are in parallel and the proper Channel is selected by the one which got Priority.
5.15
5.16
3
Yes
Runs O.K. ?
No
Data and Tape Status O.K.
No
Yes
Boot all 1's Tape to make sure no drop exists
Yes
Data and Tape Status O.K.
No
9
Manual Entry of Prelude Test

1108 BOOT / PRELOAD / SAVE TEST



Unfortunately this Test has been given about 3 different names (which are all mentioned above). This is the most complete of any of the preludes and will test quite a lot of different instructions / procedures.

Some of the Tests performed:

- Test Core Memory all ones
- Test Core Memory all zeros
- Test Core Memory addressing (Tag)
- Test J-Factors
- Store P (SLJ, LMJ)
- R1, R2 Core references

Building of PSR can be bypassed (if desired) this Tape can be put in front of DVR and will overlay itsself with JK 2, 4, 12, 14 or 15 (any)

2 different kind of Stops are used, which requires accurate Subtraction of 1 or 2 from address in P:

HJ $-B(ase) subtract 1
HJ $+1 -B(ase) subtract 2

to obtain Halt. Also keep in mind that Test Instruction might be failing!

Attempt execution in INS, CYCLE or PHASE Mode. Try to isolate failing Instruction.
Follow Boot Analysis.
Try manual entry into BOOT-TEST
(MA CLR, EXEC MODE, EXEC ABR, $p = 255_8$)

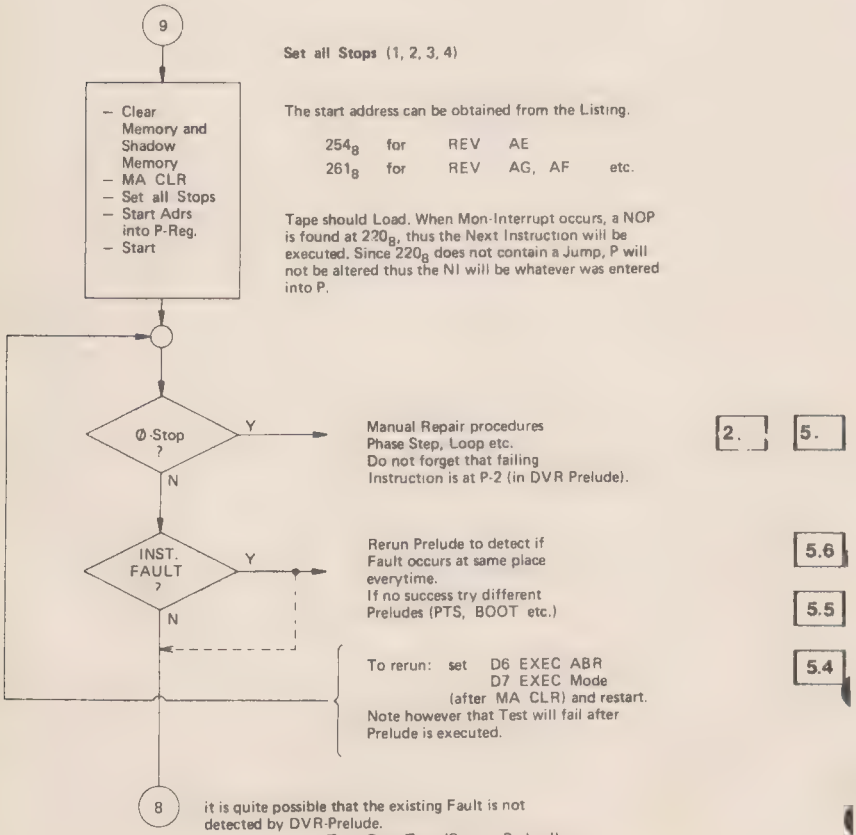
Check all ICR addresses → 13

with 000 000 000 000
777 777 777 777
252 525 252 525

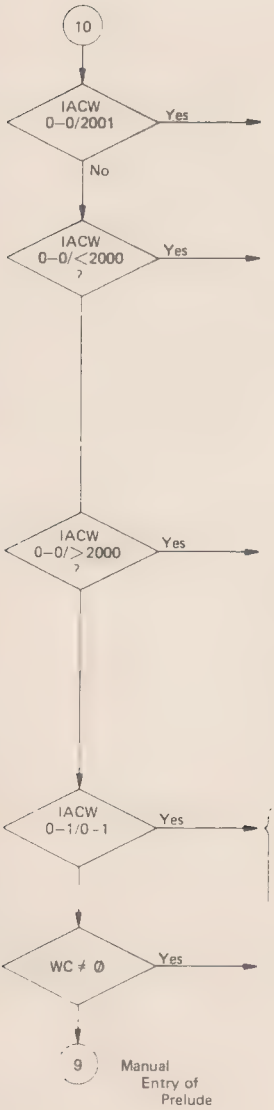
PRELUDE TEST, Manual Entry

This Test is Part of the first 2000_8 Words read during Initial Load and contains Instructions to check the Basic Functions of the CPU.
However, bear in mind that this Test assumes the proper Operation of the TEST-Instruction (thus the ARITH).
It is also possible that due to a malfunction this Test is never entered (Jump and Test Fault etc.).

The following procedure provides the Steps necessary to force entry directly to the Start of the Prelude.
Note: this procedure will only work if the MSR-Bits are equal to ØØØ.


9
Set all Stops (1, 2, 3, 4)
– Clear Memory and Shadow Memory
– MA CLR
– Set all Stops
– Start Adrs into P-Reg.
– Start
The start address can be obtained from the Listing.
254_8 for REV AE
261_8 for REV AG, AF etc.
Tape should Load. When Mon-Interrupt occurs, a NOP is found at 220_8, thus the Next Instruction will be executed. Since 220_8 does not contain a Jump, P will not be altered thus the NI will be whatever was entered into P.
Ø-Stop ?
Y
N
Manual Repair procedures
Phase Step, Loop etc.
Do not forget that failing Instruction is at P-2 (in DVR Prelude).
2.
5.
INST. FAULT ?
Y
N
Rerun Prelude to detect if Fault occurs at same place everytime.
If no success try different Preludes (PTS, BOOT etc.)
5.6
5.5
To rerun: set D6 EXEC ABR
D7 EXEC Mode
(after MA CLR) and restart.
Note however that Test will fail after Prelude is executed.
5.4
8
it is quite possible that the existing Fault is not detected by DVR-Prelude.
Try Boot-Test (Save or Preload)
also PTS-Prelude

ACCESS CONTROL WORD ANALYSIS
(INITIAL LOAD IND. OFF)



MP-Switch (on Maintenance Panel) activated but should only be used if Booting via Input-Output Controller

Bit dropped in address or word cut (V,W)
If ICR-error: $ZØ_{lower,\ upper}$
Sense Amp
ICR-Circuit (IC)
Address Read or Write
If no error: $ZØ_{lower}$
ICR → W1 drop
IXA 1 → ZØ
+1 → W3
Index addes etc.
drop in W2 or W3

3. 5.5 5.4 (11)

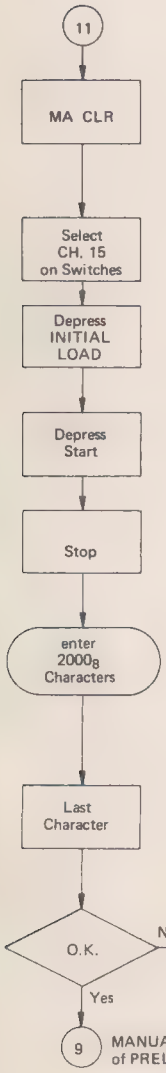
Bit picked in Address or W-Portion of AWC
If ICR error: $ZØ_{lower,\ upper}$
Sense Amp
ICR-Circuit (IC)
Address Read or Write
If no error: $ZØ_{lower}$
ICR → W1 p.u.
IXA1 → ZØ
+1 → W3
Index Adder etc.
pick in W2 or W3

3. 5.5 5.4 (11)

Wrong Access Control Register accessed, resulting in Initial WC = Ø, causing Mon Intrpt immediatly.
ICR → W1, W2 fails etc.
Initial Load failed etc.

Indicates that MON Intrpt occurred even though the Word Count was not zero. Check clear IN ACTIVE and MON FF etc.

BOOT FROM CHANNEL 15



Initial Load assumes working from Master cleared state.
Check vital Registers to see if CLR is performed correctly.

2.1.5

Should be Loaded → CSR → F1a when initial Load initiated.

V7, V8 set: 2^{35}, (also $2^{0,1,2}$ if MP)
XØ set: 2^{28}, MSR → $2^{15,16,17}$
(2^{13} instead of 2^{28} if MP switch)
Note: if Booting from CH 15 MP-switch must be down!

Access Control word should be written into ICR (57_8) observe at F3
IN ACTIVE on Channel # 15.
MON FF set (cannot be observed)

Stop again, to avoid INS Fault when Monitor Interrupt occurs.
(only I Char. is entered into each word)

each Character entered should:
Increment V ($ZØ_L$)
Decrement W ($ZØ_U$)
Data → Memory 1 2^0–2^5 (if MSR = Ø)
Address at Memory 1 (if MSR = Ø)

When the Last Character is entered, a IN MON INTRPT is initiated to Location 220_8.

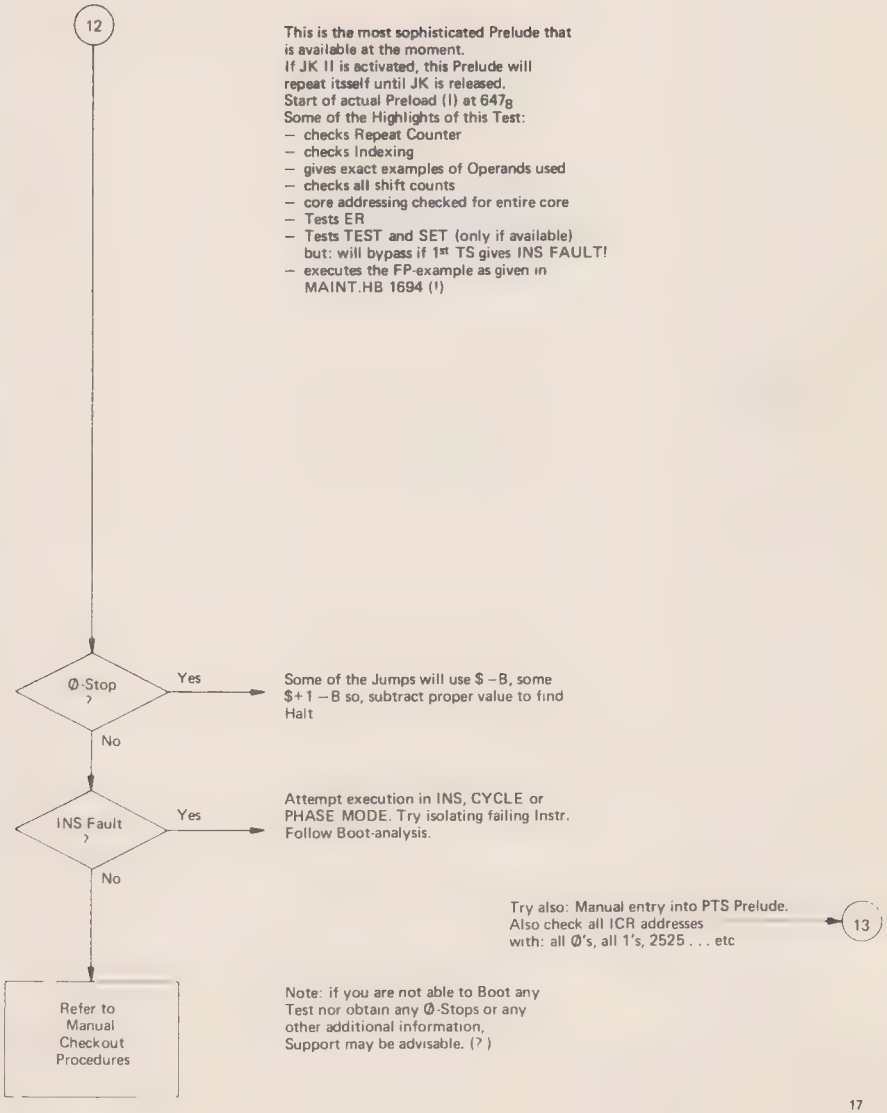
Manual repair Procedure should Lead to Fix
Also refer to BOOT Analysis

3. 5.5 5.15 5.16

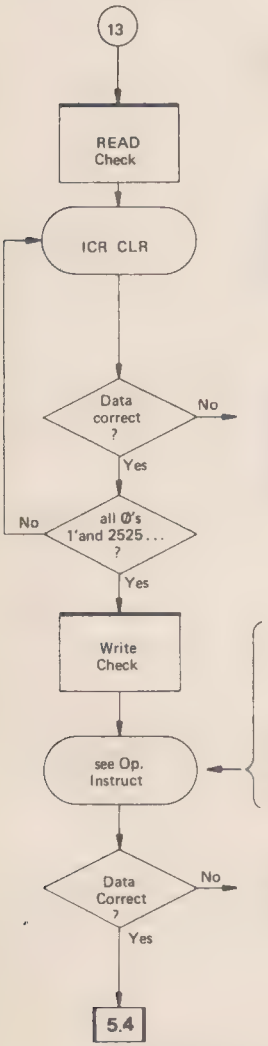
7

Also try:
different values of G in Access Control Word.
(observe V-portion)

PTS Prelude (Bootable)


12
This is the most sophisticated Prelude that is available at the moment.
If JK 11 is activated, this Prelude will repeat itsself until JK is released.
Start of actual Preload (!) at 647$_8$
Some of the Highlights of this Test:
– checks Repeat Counter
– checks Indexing
– gives exact examples of Operands used
– checks all shift counts
– core addressing checked for entire core
– Tests ER
– Tests TEST and SET (only if available) but: will bypass if 1st TS gives INS FAULT!
– executes the FP-example as given in MAINT.HB 1694 (!)
Ø-Stop ?
Yes
Some of the Jumps will use $ – B, some $+1 – B so, subtract proper value to find Halt
No
INS Fault ?
Yes
Attempt execution in INS, CYCLE or PHASE MODE. Try isolating failing Instr. Follow Boot-analysis.
No
Try also: Manual entry into PTS Prelude.
Also check all ICR addresses with: all Ø's, all 1's, 2525 . . . etc
13
Refer to Manual Checkout Procedures
Note: if you are not able to Boot any Test nor obtain any Ø-Stops or any other additional information, Support may be advisable. (?)

Verify all ICR-addresses



It might be helpful to establish whether all ICR-addresses can be accessed with READ and WRITE!

Clear ICR with Ø's, then 1's, then 2525 . . . then step through with a Load Instruction, starting at 002, using Index-Modification of Register 001 (x 1). Note: Observe R1, R2 to make sure that Increment works satisfactorily.

Apply Manual Repair Procedures. Also note that all 1's will be read if Read-address is missing (no parity error)
Note: Dual Selections will not be detected!
Suspect Read Driver and Base Switch. Do not remove this card with power on!
ICR CLR might not write into Location!

Clear ICR with all Ones, Manually write all zeroes into all Locations (not with ICR clear!!!) Reread each Location with method described above.
Note: Dual Selection will not be detected!

Manual repair procedures
Suspect Write Driver and Base Switch
Do not remove this card with DC on!

also attempt Execution of ICR Tag routine which might detect some Dual Selections

1.5 INITIAL LOAD TROUBLE SHOOTING ON 1100/10 OR /20

The following information is an addition and a modification of Section 1. It is based on CCDTS as used on the 1100/10/20. The procedure and flow chart can be used as well on the 1106 or 1108 because hardware problems will show up in always the same manner. This section will not cover the specialities of DVR as e.g. Jump Key selection. Basically the CCDTS will work in the same way as DVR does with Jump Key 15.

Note 1: Prepare System for Test

- Disable Day Clock and Real Time Clock
- Memories on line
- Tape Subsystem on line
- Select Servo in Control Unit or MSA

Note 2: Initial Load (Bootstrap)

- Press System Clear
- Press I/O Clear (Subsystem Clear)
- Select first Memory available (normally 0)
- Select Boot Channel on Initial Load switches
- Press Initial Load
- Press Run

Note 3: Boot from FH Drum or Console

- Booting from these devices allows phase stepping through the I/O
- Boot from 40/3 Console

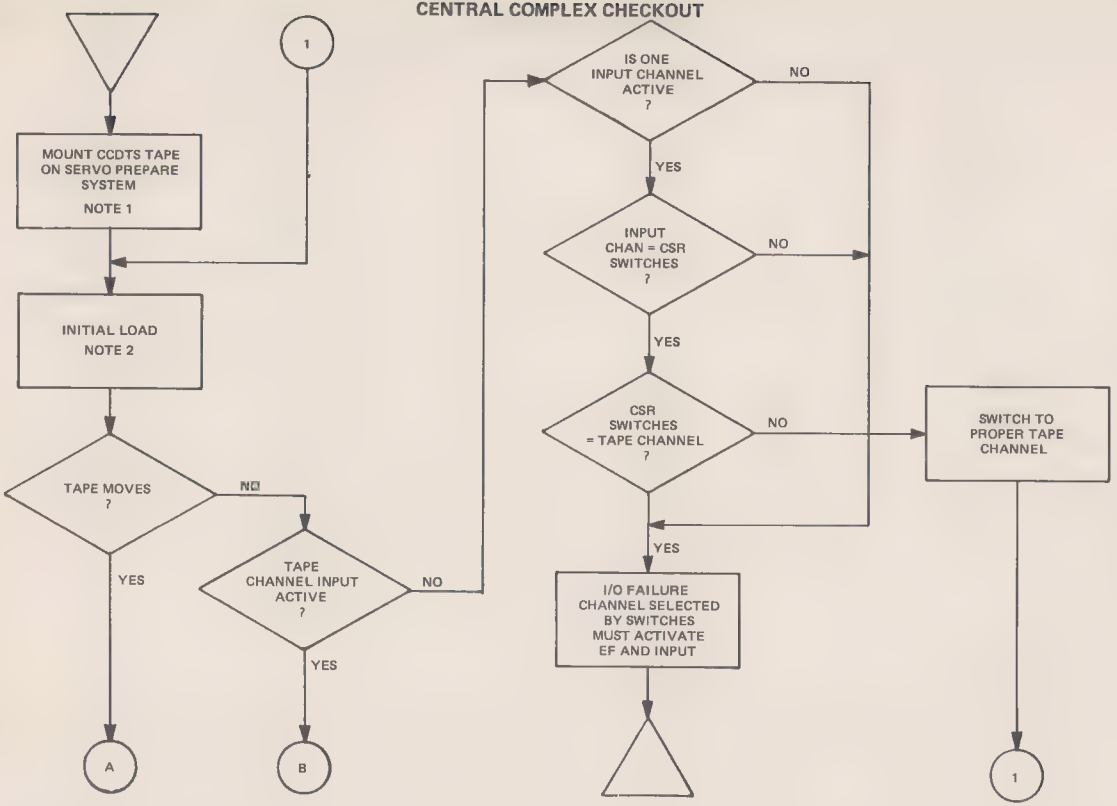
  Type message and press Transmit. The last character transferred is the one under the cursor
- Boot from 4009 Console

  Pressing a key will transfer one character

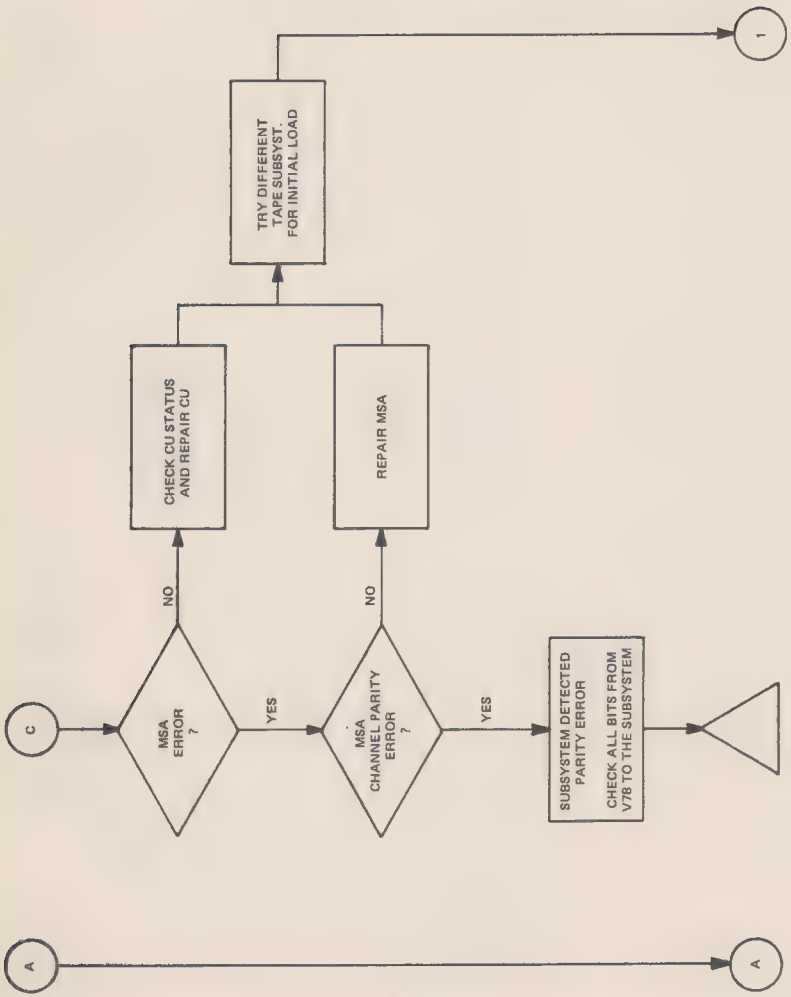
Note 4: Channel Parity Error

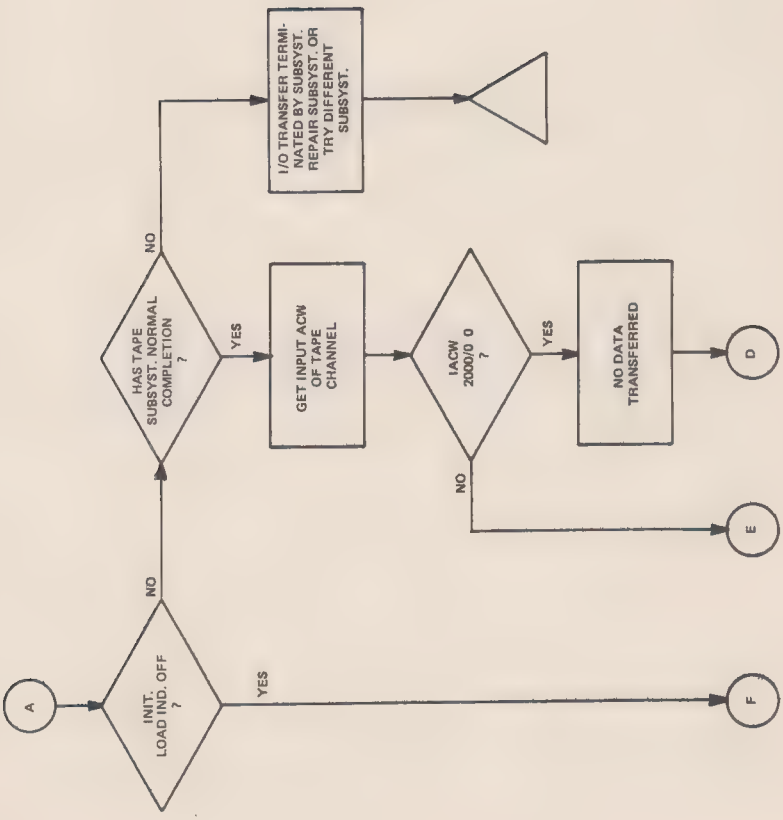
This is detected in the Memory, generating a Memory Parity Error. To determine if the fault was caused by a wrong I/O transfer, put the Memory in Stop On Error and check the CPU Maintenance Panel if Main Control or I/O was running while the fault occurred.

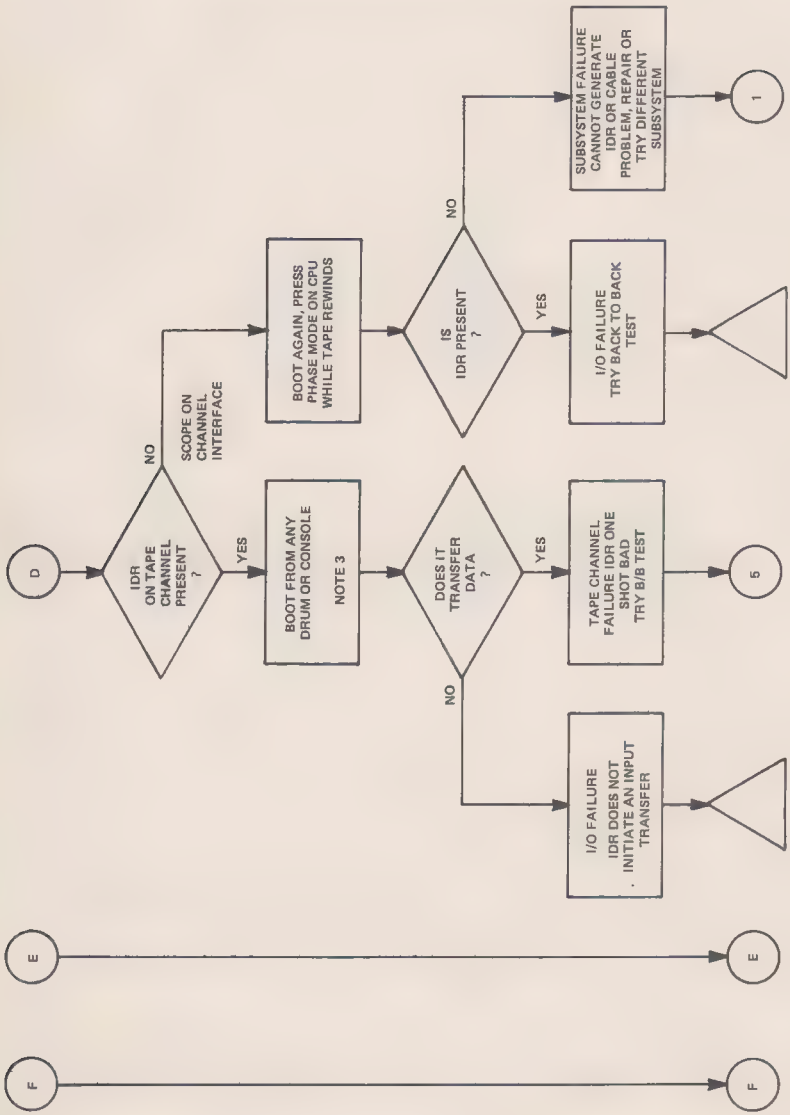
CENTRAL COMPLEX CHECKOUT



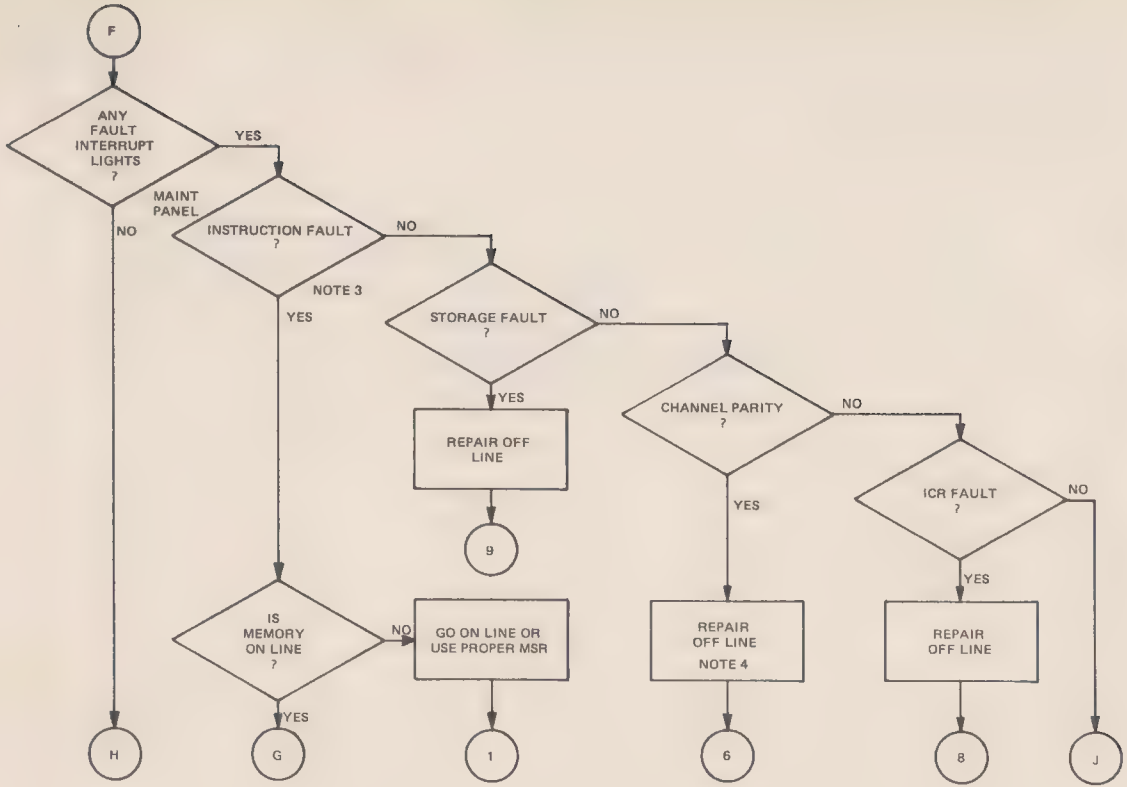
A
B
C
TAPE SUBSYST. RECEIVED EF ?
NO
YES
SUBSYSTEM GENERATED AN EI ?
NO
YES
PROPER TAPE UNIT SELECT ?
NO
YES
SUBSYSTEM HANGS. REPAIR OR TRY DIFFERENT SUBSYSTEM
1
CORRECT TAPE UNIT SELECT
1
IS SUBSYSTEM ON LINE ?
NO
YES
GO ON LINE
1
ON MSA IS PROPER INTERFACE CONNECTED ?
NO
YES
MANUAL CLEAR MSA AND CU
TRY AGAIN
2nd TRY ?
NO
YES
SO THE SUBSYSTEM RECEIVED THE EF GO BACK I
EF SIGNAL TO SUBSYST LOST STEP IN PHASE MODE THROU INIT LOAD TO GENERATE AND TRACE THE EF
1

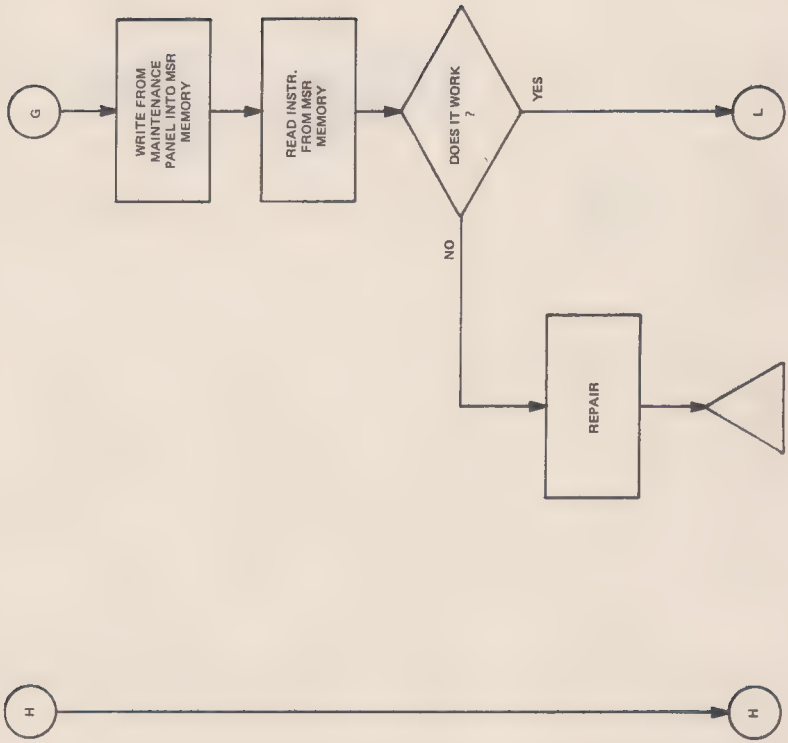

A
C
MSA ERROR ?
YES
NO
CHECK CU STATUS AND REPAIR CU
MSA CHANNEL PARITY ERROR ?
YES
NO
REPAIR MSA
SUBSYSTEM DETECTED PARITY ERROR
CHECK ALL BITS FROM V78 TO THE SUBSYSTEM
TRY DIFFERENT TAPE SUBSYST. FOR INITIAL LOAD
1
A

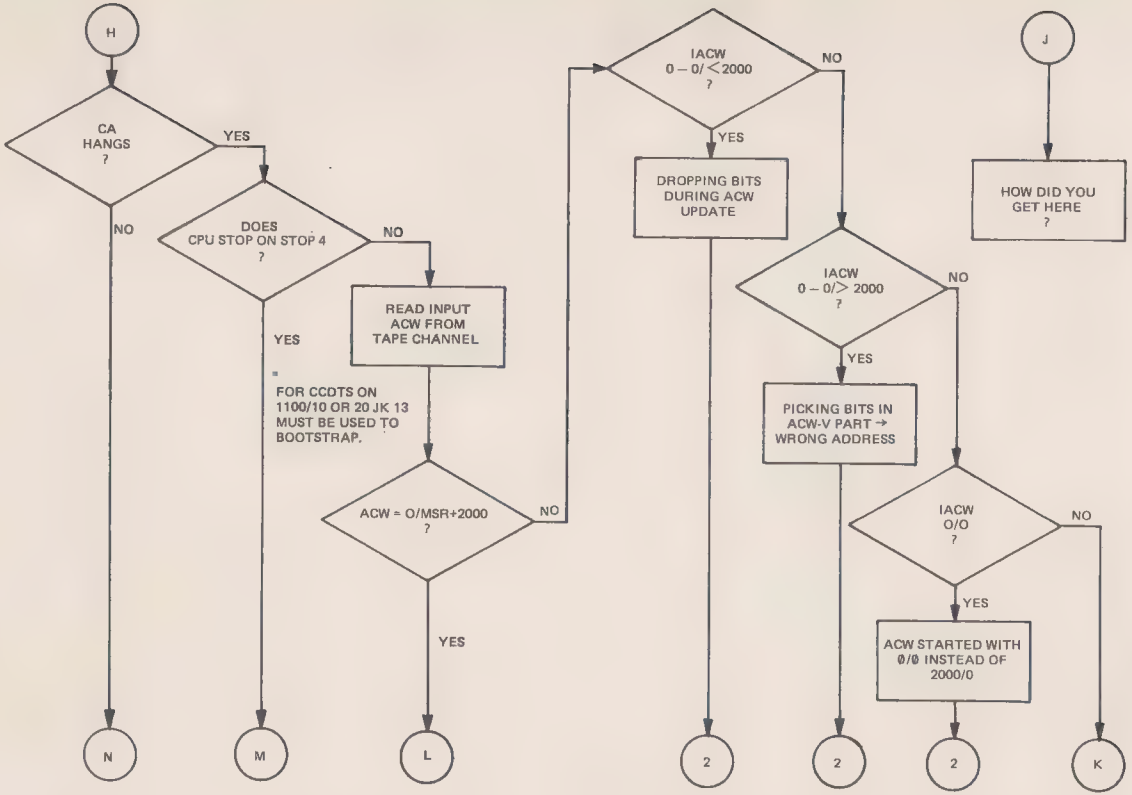

A
INIT. LOAD IND. OFF ?
NO
YES
HAS TAPE SUBSYST. NORMAL COMPLETION ?
NO
YES
I/O TRANSFER TERMINATED BY SUBSYST. REPAIR SUBSYST. OR TRY DIFFERENT SUBSYST.
GET INPUT ACW OF TAPE CHANNEL
IACW 2000/0 0 ?
YES
NO
NO DATA TRANSFERRED
D
E
F

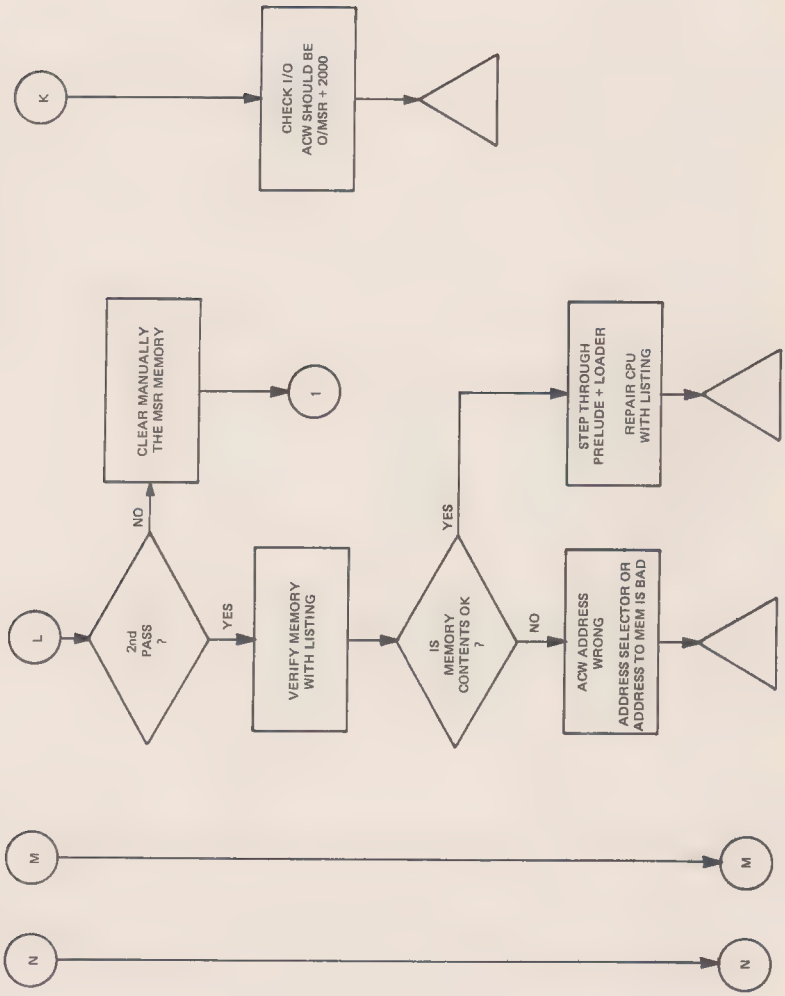

D
IDR ON TAPE CHANNEL PRESENT ?
NO
SCOPE ON CHANNEL INTERFACE
YES
BOOT FROM ANY DRUM OR CONSOLE
NOTE 3
BOOT AGAIN, PRESS PHASE MODE ON CPU WHILE TAPE REWINDS
IS IDR PRESENT ?
NO
SUBSYSTEM FAILURE CANNOT GENERATE IDR OR CABLE PROBLEM, REPAIR OR TRY DIFFERENT SUBSYSTEM
1
YES
I/O FAILURE TRY BACK TO BACK TEST
DOES IT TRANSFER DATA ?
YES
TAPE CHANNEL FAILURE IDR ONE SHOT BAD TRY B/B TEST
5
NO
I/O FAILURE IDR DOES NOT . INITIATE AN INPUT TRANSFER
E
E
F
F

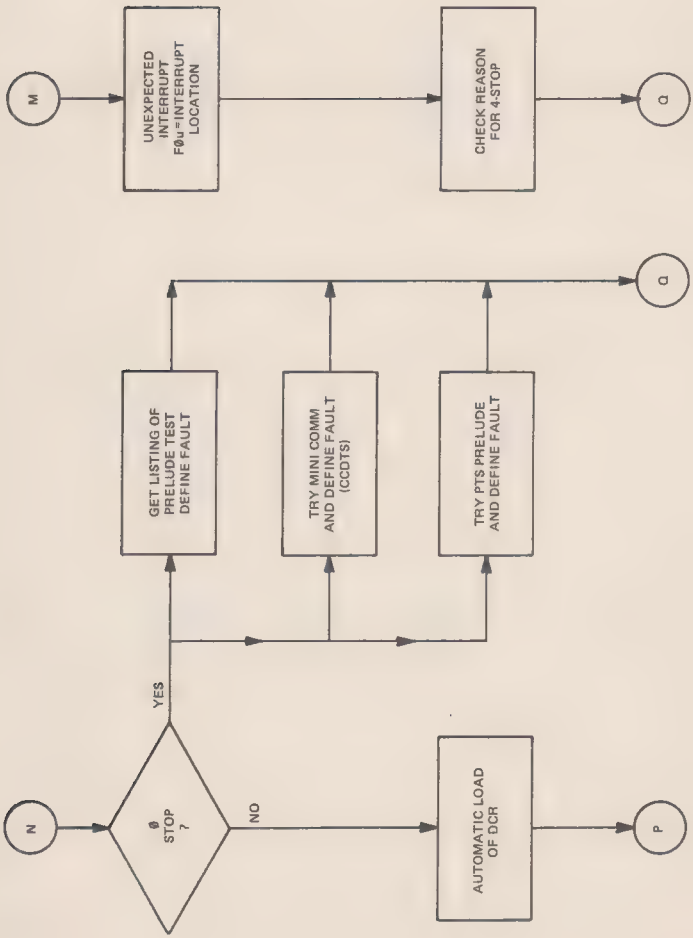
E
IACW 1777/0 - 01 ?
YES
NO
TRANSFERRED ONLY 1 WORD
CHECK IA TO THE SUBSYST. LINE DRIVE CABLE, INPUT AMPLIFIER
IACW 0 - 0/02000 ?
YES
NO
IOAU MISSED GENERATING THE MONITOR INTERRUPT
CHECK IF INPUT STILL ACTIVE, InMon STILL SET
IACW >0/<2000 ADD W+V -2000
YES
NO
TAPE FAULT BOOT BLOCK IS SMALLER THAN 2000 WORDS
TRY DIFFERENT TAPE OR MANUAL WRITTEN TAPE
IACW 0/>2000 ADD W+V =>2000
YES
NO
I/O FAILURE ACW UPDATING BAD BIT PICKING
MANUAL B/B TEST PROCEDURE
IACW XXXXX/10777 ?
YES
NO
.W = 0 NOT DETECTED OR ACW IS PICKING BITS
CHECK WORD COUNT = 0 DETECTION + ACW UPDATING
VERY FUNNY - YOU SHOULD NOT ARRIVE HERE
FOR I/O TRY MANUAL CHECKOUT
F
F


F
ANY FAULT INTERRUPT LIGHTS ?
YES
NO
MAINT PANEL
INSTRUCTION FAULT ?
NOTE 3
NO
YES
STORAGE FAULT ?
NO
YES
REPAIR OFF LINE
9
CHANNEL PARITY ?
NO
YES
ICR FAULT ?
NO
YES
IS MEMORY ON LINE ?
NO
YES
GO ON LINE OR USE PROPER MSR
REPAIR OFF LINE NOTE 4
REPAIR OFF LINE
H
G
1
6
8
J

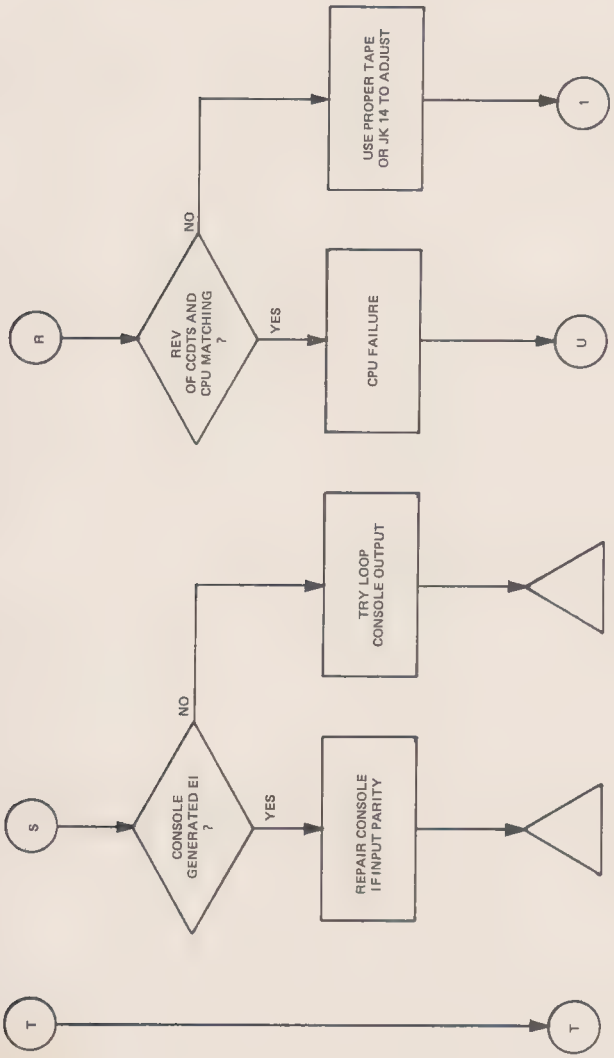

G
WRITE FROM MAINTENANCE PANEL INTO MSR MEMORY
READ INSTR. FROM MSR MEMORY
DOES IT WORK ?
YES
L
NO
REPAIR
H
H

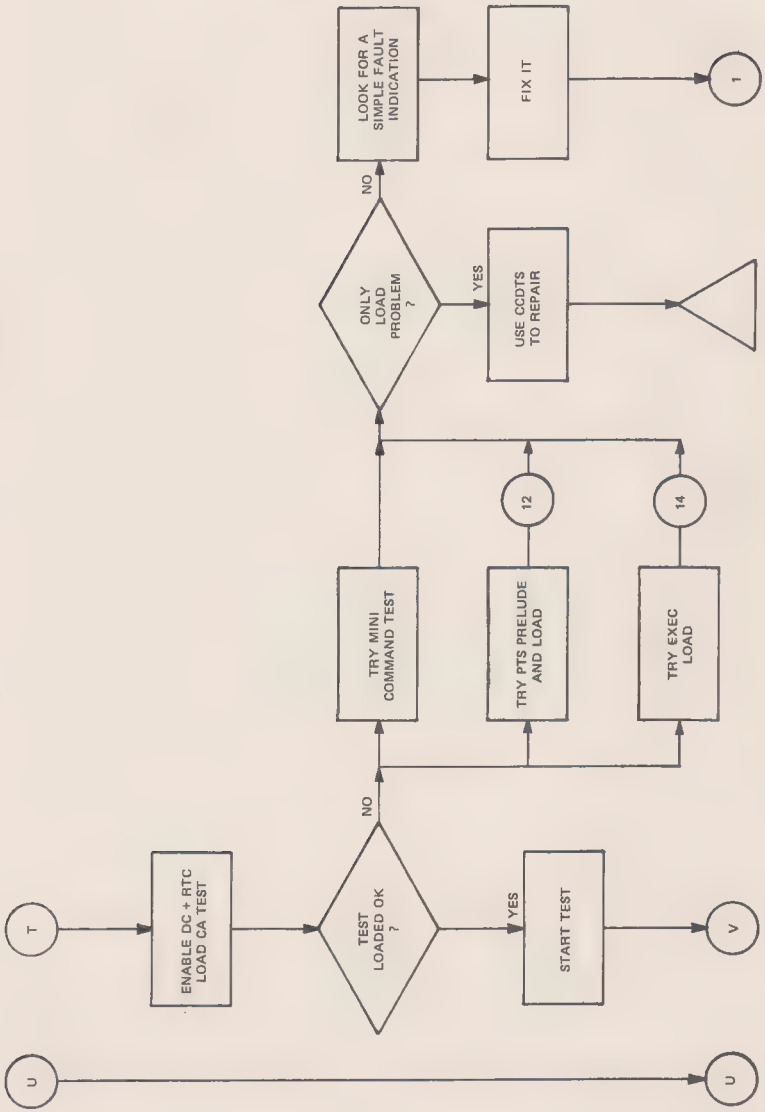

H
CA HANGS ?
YES
NO
DOES CPU STOP ON STOP 4 ?
NO
YES
READ INPUT ACW FROM TAPE CHANNEL
FOR CCDTS ON 1100/10 OR 20 JK 13 MUST BE USED TO BOOTSTRAP.
ACW = O/MSR+2000 ?
NO
YES
N
M
L
IACW 0 – 0/ < 2000 ?
NO
YES
DROPPING BITS DURING ACW UPDATE
IACW 0 – 0/ > 2000 ?
NO
YES
PICKING BITS IN ACW-V PART → WRONG ADDRESS
IACW O/O ?
NO
YES
ACW STARTED WITH Ø/Ø INSTEAD OF 2000/0
J
HOW DID YOU GET HERE ?
2
2
2
K

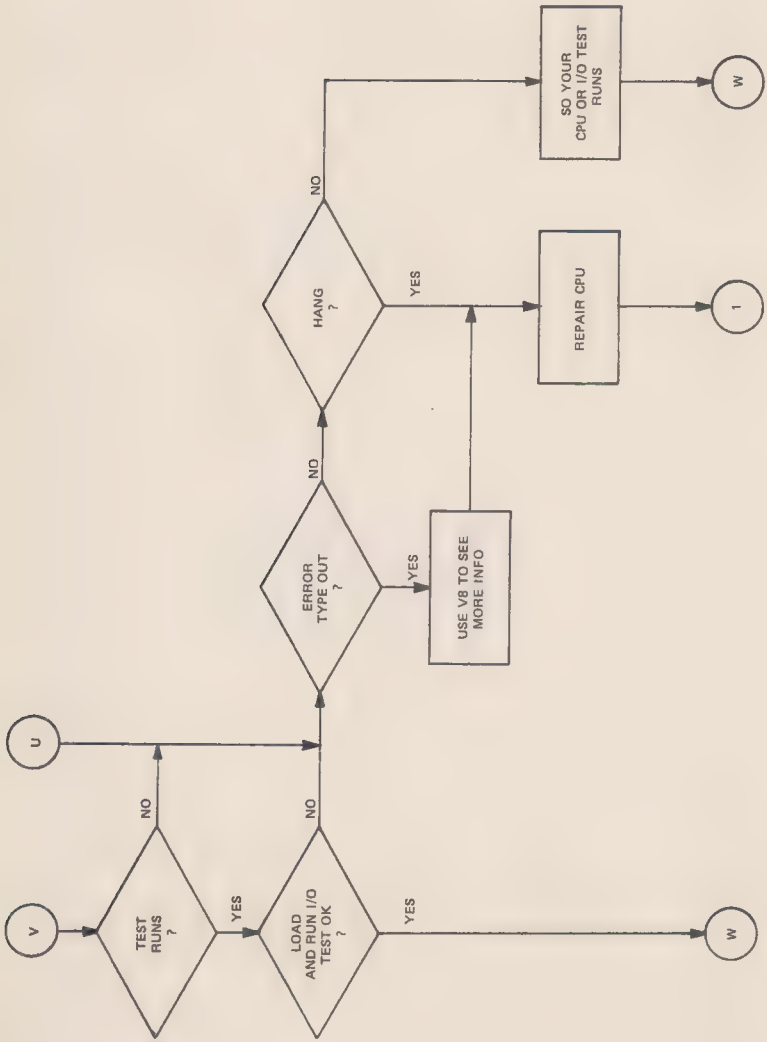

K
CHECK I/O
ACW SHOULD BE
O/MSR + 2000
L
2nd
PASS
?
NO
CLEAR MANUALLY
THE MSR MEMORY
I
YES
VERIFY MEMORY
WITH LISTING
IS
MEMORY
CONTENTS OK
?
YES
STEP THROUGH
PRELUDE + LOADER
REPAIR CPU
WITH LISTING
NO
ACW ADDRESS
WRONG
ADDRESS SELECTOR OR
ADDRESS TO MEM IS BAD
M
M
N
N


M
UNEXPECTED INTERRUPT FØu=INTERRUPT LOCATION
CHECK REASON FOR 4-STOP
Q
Q
GET LISTING OF PRELUDE TEST DEFINE FAULT
TRY MINI COMM AND DEFINE FAULT (CCDTS)
TRY PTS PRELUDE AND DEFINE FAULT
YES
N
Ø STOP ?
NO
AUTOMATIC LOAD OF DCR
P

Q
FIX FAULT
1
P
JK 12 SET ?
NO
YES
DCR LOAD OK ?
NO
YES
SAME AS Ø STOP
N
AUTOMATIC LOAD OF THE UNCONDITIONAL CA TEST
CONSOLE TYPE OUT OK ?
NO
YES
ERROR JK 12 NOT SET ?
NO
YES
T
R
CONSOLE OUTPUT CHANNEL ACTIVE ?
NO
YES
2nd PASS ?
NO
YES
S
CLEAR CONSOLE BY PRESSING U100 WAIT
1
ANY OTHER CHANNEL ACTIVE (CSR) ?
NO
YES
TRY BOOT WITH JK 8
1
GO THROUGH DCR
2

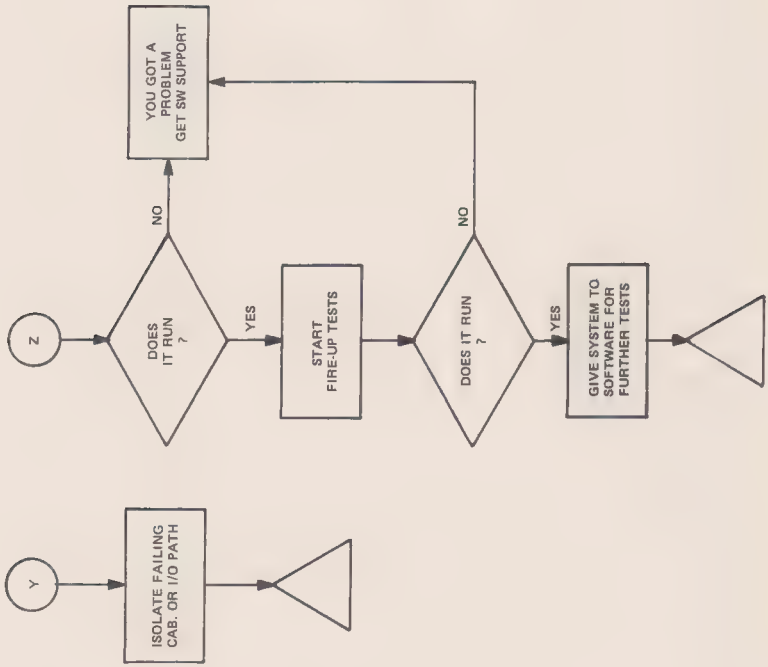

R
REV OF CCDTS AND CPU MATCHING ?
NO
YES
USE PROPER TAPE OR JK 14 TO ADJUST
I
CPU FAILURE
U
S
CONSOLE GENERATED EI ?
NO
YES
TRY LOOP CONSOLE OUTPUT
REPAIR CONSOLE I F INPUT PARITY
T
T


T
ENABLE DC + RTC
LOAD CA TEST
TEST
LOADED OK
?
NO
YES
START TEST
V
U
U
TRY MINI
COMMAND TEST
TRY PTS PRELUDE
AND LOAD
12
TRY EXEC
LOAD
14
ONLY
LOAD
PROBLEM
?
NO
YES
USE CCDTS
TO REPAIR
LOOK FOR A
SIMPLE FAULT
INDICATION
FIX IT
1


V
TEST RUNS ?
NO
YES
U
LOAD AND RUN I/O TEST OK ?
NO
YES
W
ERROR TYPE OUT ?
NO
YES
USE V8 TO SEE MORE INFO
HANG ?
NO
YES
REPAIR CPU
1
SO YOUR CPU OR I/O TEST RUNS
W

W
LOAD AND RUN MEM TESTS OK ?
NO
YES
X
MEM FAULT INTERRUPT ?
NO
YES
READ DATA PARITY ERROR ?
NO
YES
REPAIR MEMORY OFF LINE
1
WRITE DATA, WRITE CONTROL OR ADDRESS PARITY ERROR
CAN FAULT BE REPRODUCED FROM CPU ?
NO
YES
REPAIR MEMORY OFF LINE
1
REPAIR ON LINE USING CPU MAINT PANEL
1
DOES CAU OR IOAU HANG ?
NO
YES
USE HANG PROCEDURE TO DEFINE FAULT
REPAIR
1

X
TEST CA + I/O WITH VARIOUS MSR's
ALL TESTS OK ?
NO
YES
ISOLATE AND REPAIR FAILING CABINET
IS A CENTRAL COMPLEX FAULT SUSPECT ?
NO
YES
RUN CA + I/O MARGIN TESTS
LOAD PTS
TEST ALL CU's ON ALL PATHS WITH ALL TESTS
FAILURE ?
NO
YES
Y
LOAD 1100 OS
Z


Z
DOES IT RUN ?
NO
YOU GOT A PROBLEM GET SW SUPPORT
YES
START FIRE-UP TESTS
DOES IT RUN ?
NO
YES
GIVE SYSTEM TO SOFTWARE FOR FURTHER TESTS
Y
ISOLATE FAILING CAB. OR I/O PATH

EUROPEAN EDUCATION AND TRAINING CENTER

SECTION 2

STATIC CHECKS MAIN CONTROL

CONTENTS LIST

		Page No.
2.1	READ/WRITE MEMORY AND ICR	1
2.1.1	How to Write into Memory	1
2.1.2	How to Write into ICR	1
2.1.3	How to Read Memory/ICR	1
2.1.4	How to Access Consecutive Locations	2
2.1.5	Master – I/O Clear	2
2.2	MANUAL CHECKOUT PROCEDURES	5
2.2.1	P–Register, Updating, P → S	5
2.3	RØ REGISTER	6
2.4	R1 & R2 CHECKOUT PROCEDURE	8
2.5	R1 OR R2 → RSEL → W4 → P	9
2.6	MAIN CONTROL	10

2. STATIC CHECKS MAIN CONTROL

2.1 READ/WRITE MEMORY AND ICR

2.1.1 How to Write into Memory

- MA CLR
- all disables
- enter Ø5—Instruction (SZ) into FØ
- INS MODE
- Enter Data into XØ (ARITH)
- Start
- Check Memory: Data should appear in MDR, all Write Groups should be ON.

Note: This will not work in Cycle or Phase Mode!

2.1.2 How to Write into ICR

- MA CLR
- all disables
- enter Ø5—Instruction (SZ) into FØ
- INS MODE
- Enter Data into AØ (ARITH)
- Start
- Check: Data should have appeared in ZØ

Note: This will not work in Phase Mode!

OR

2.1.2.1
- set data into AØ
- press ICR Clear

This will write the data into all ICR locations

2.1.3 How to Read Memory/ICR

- MA CLR
- all disables except ARITH
- enter Load Instruction into FØ

```
        f  j  a  x  hi  u
        10 00 00 00 0  Address
```

(if more than the lowest 65K are to be accessed, an Index-Register must be used!)

- INS MODE
- Start, Contents of Location specified by Ueff are to be seen in AØ of Arithmetic.

Note: To display the contents of a Location in Shadow Memory, a special procedure with BI, BD and address Overflow is necessary.

e.g. FØu = 001XXX
BI = BD = 777
R1= R2 = 000XXX = < 200 Hidden Memory holder
R0 = 1XXX = > 200 therefore Core Read u
XXX address = < 0200

This produces an address above Ø but less than 200_8. However, Ueff is greater than 200_8 so this allows access to Shadow Memory instead of ICR.

2.1.4 How to Access Consecutive Locations

in ICR or Core Memory

- MA CLR, ICR CLR
- $000^{Xi}001_8$ $SSS^{Xm}SSS_8$ into AØ of Arith where SSS equal Start-address of Read or Write
- ICR Clear (this will write the Data into all Locations)

Read Procedure

- INS MODE, all DISABLES except ARITH
- enter following Instruction into FØ:

```
f  j  a  x  hi   u
10 00 00 XX 2 000000
```

where XX any of 15 Index-Registers

- To restart, use a different X-Reg.

Write Procedure:

- INS MODE, all DISABLES
- enter following Instruction into FØ

```
f  j  a  x  hi     u
05 00 00 XX 2 000 000
```

where xx any of 15 Index-Registers
- To restart, use a different X-Reg.

2.1.5 Master – I/O Clear

Master clear check

Before setting any indicators, put the rate control in phase step mode. Then start at the bottom of the panel and set a row of indicators (NOTE: Do not set the following indicators: $SØ_R$ reg., $SØ_W$ reg., BØ bit 35, ICR WR 37, 36, ICR RD 37, 36, Cycle II, Clk., Clk. Stop I, Clk. Stop II, and DIST 1, 2, 3, 4). Master Clear the computer and see that the bits clear. Continue this procedure, moving up through the entire maintenance panel. The following registers and designators will not master clear; W1, W2, W3, W4, W5, RØ, R1, R2, Brkpt., QØ, Q1, Q2, Q3, BØ, B2, A1, B1, A3, B3, ZØ V7, V8, CØ C1, C2, CP, V6, and V4; manually clear all registers and then continue. F1 reg. A position: A2 register bits 0 & 1: V4 register; Buffer 36, 37, Borrow S1, D1, FEBI, EAB, Neg ADD, ACONFL, NI, RØ – EØ, Core RD U, ICR RDU, ICR WRU, W4=0, EI DIS, INDR, ABORT', HS JUMP, HS JP; ARITH JUMP, R1, ARITH MEM, ARITH COMPL, REQ O.S., BLK RØ Designators, and TØ LAST.

1108 CONTROL DESIGNATORS

	Page		Page
ICR Read u	1	P, R1, R2 → S Mem 1	27A
ICR Write u	1	P, R1, R2 → S Mem 2	27B
Core Read u	2	P, R1, R2 → S Mem 3	28A
R0 → X0	2	P, R1, R2 → S Mem 4	28B
NI NI'	2	R1 wait Odd and Even	28C
IFR Des	2	R2 wait Odd and Even	28D
Pass 2, Pass 2'	3	P wait Odd and Even	28E
Indirect	3	Arith Mem	30
EI Disable	3	Arith Comp. → Mem	30
Abort Abort'	4	Store Des, F2	34
BD Des	4	3T Write	34
RI Des	4	ICR Wr Up & Lower	37
MSR	4	TO lock out	43
W4 = Zero	5	OP lock out	43
RPT in Progress	5	Phase & Cycle	129
RPT Set up	5	Inst + Normal	129
Arith Jump	6	Clock Stop I & II	129
HS Jump Des	6	Phase 1, 2, 3, 4	130
Prog Interrupt Dis	7	Cycle I & II	130
Interrupt Interrupt'	7	Dist 1, 2, 3, 4	130
Sync I and Sync II	8	Phase Repeat	130
Exec Mode	8	Illegal Inst	132
Exec ABR	8	Mem Par. 1, 2, 3, 4	132
Block DP Underflow	8	ICR Par	132
1107 Compat	8	Prog Alarm	132
Write Protection	8	Audio	132
Guard Mode	8	100 CPS Osc	132
I/O Reg One Shot	9	Select BRKPT. Stop & R	135
Time Out One Shot	9	Stop 1, 2, 3, 4	134
A Conflict	10	Clock Osc	138
B Conflict	11	ICR Data 36 & 37	173
Guard Mode Fault	16	1107 FP Zero	8
Exec Return	16	TO last	16
Fault Des	17	SOR Fault	40A
		Block RO	40B

EUROPEAN EDUCATION AND TRAINING CENTER

1108 ARITH DESIGNATORS

	Page
Norm → C3	203
Const Des	203
Neg Add	204
EAB	204
Ext Seq	232
Sign I & II	232
Shift	232
Ovfl	233
Underfl	233
Fault	233
Mant Zero	233
Carry	234
Ovfl	234
Skip I & II	234
Jump	235
Divide Start	238
Carry S1, S2	240
FEB 1 & 2	240
Carry D1, D2	241
Borrow S1, S2	242
Borrow D1, D2	243
Buffer 36, 37	244
X2 Des I	255
X2 Des II	256
X2 Des III	256
X2 Des IV	256
X2 Des V	256
X3 Des	257
X4 Des	257
X6 Des	258
X7 Des	259
Strip	264

1108 I/O DESIGNATORS

	Page
In & Out Mon (ESI)	601
EI (ESI)	601
Power Loss & Parity Interr.	601
RTC Dec	601
In & Out	601
Sync I & II	602
Func. Mon	602
In & Out Mon (ISI)	602
EI (ISI)	602
RTC Interrupt	602
Interrupt L.O.	605
Interrupt Req	605
Valid F.P.	605
ESI-I (OP)	611
RC-1	612
WC-1	612
ESI Act	612
RW-1	612
Initial Load	628
I/O Abort A & C	628
Init Load B & D	628
Ack Control	614
Ack 0–7 Out, in, EF	613
Ack 0–8 Out, in, EF	613

1108 DISABLES

	Page
Core Mem	133
Day Clock	618
Real Time Clock	618
Z → V7 Z → V8	623
Advance P	133
Arith Interrupt	133
Mem Fault Interrupt	133
Read NI	133
Write A	133
Arith Seq	133

1108 CONTROLS

Alarm Reset	132
Fault Reset	132
I/O Clear	131
ICR Clear	131
Master Clear	131
Start	128
Stop	128

1108 MODES

	Page
Real Time	133
Parity Stop	128

1108 SWITCHES

Init Load, Chan 0, 1	616
Init Load, Chan 2, 3	617
Compatible Mode Chan	604
En Power loss	618
Dis D.P.	133
Multi Proc	133
EI Bias	133
Last Address	133
ESI Chan	679

2.2 MANUAL CHECKOUT PROCEDURES

2.2.1 P-Register, updating, P → S

(P, P → W4, W4, PADD)

2.2.1.1 P → W4 → PADD → P (under Control of T3)

- MA CLR
- all disables except ADV P
- FØ = 74 06 00 00 0 0 ——— 0
- Run: — all bits should be visible at W4 and P
  - if INS Mode: W4 = P – 1
  - if Adder drops bit, next Pos. will never be reached
  - pick is very hard to see static!

2.2.1.2 P → S

- set up as under 2.2.1.1.
  - Check all Memory Modules: all bits should arrive!
- set up the following
- all disables, FØ = Load-Inst. Ueff $< 200_8$
- select P as follows:

	Non-Interleave			Interleave			
Memory	2^{17}	2^{16}	2^{15}-2^0	2^{17}	2^{16}	2^{15}-2^1	2^0
M1 even	0	0	0	0	0	0	0
M2 even	0	1	0	1	0	0	0
M3 even	1	0	0	0	1	0	0
M4 even	1	1	0	1	1	0	0
M1 odd	-	-	-	0	0	0	1
M2 odd	-	-	-	1	0	0	1
M3 odd	-	-	-	0	1	0	1
M4 odd	-	-	-	1	1	0	1

- Observe: when Module is selected, all Address Lines should be zero, otherwise a pick exists.

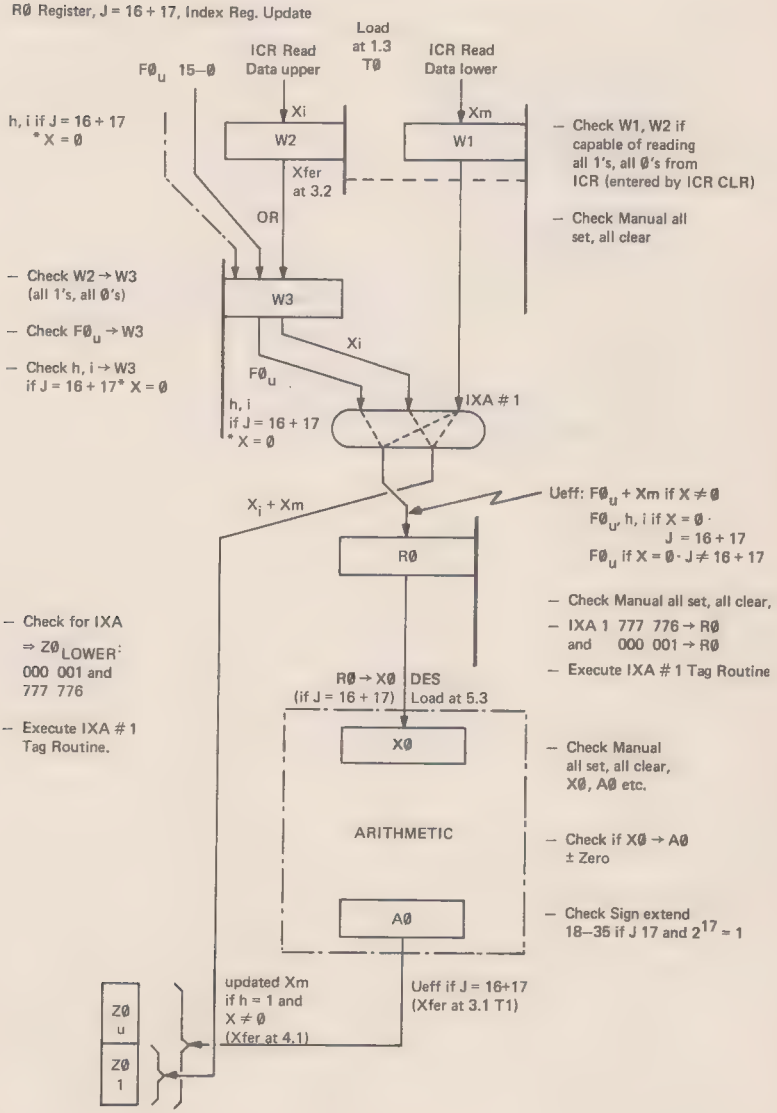
2.3 RØ REGISTER

all disables

Static checkout: Set and clear W1, 2 & 3 RØ all bits

- Load IXR1 with xi = 1 xm = 0
- FØ = 10 00 00 01 2 0———0
- Run
- check RØ should show all bits (counting up)
- observing RØ, add bits in FØu 1 by 1. RØ should still show all bits
- clear FØu
- Repeat with xi = -1
- FØ = 1Ø 16 00 00 XX X———X
- Add bit on u 0 - 16 h i, the same bits should appear in RØ
- Execute J-Factors Routine 5.4
- Execute Index Adder # 1 Routine 5.4a

Note: Bit picking is very difficult to see.


RØ Register, J = 16 + 17, Index Reg. Update
Load at 1.3 TØ
ICR Read Data upper
ICR Read Data lower
FØu 15–Ø
Xi
Xm
W2
W1
Xfer at 3.2
OR
W3
h, i if J = 16 + 17 * X = Ø
– Check W1, W2 if capable of reading all 1's, all Ø's from ICR (entered by ICR CLR)
– Check Manual all set, all clear
– Check W2 → W3 (all 1's, all Ø's)
– Check FØu → W3
– Check h, i → W3 if J = 16 + 17* X = Ø
FØu
h, i if J = 16 + 17 * X = Ø
IXA # 1
Xi + Xm
Ueff: FØu + Xm if X ≠ Ø
FØu, h, i if X = Ø · J = 16 + 17
FØu if X = Ø · J ≠ 16 + 17
RØ
– Check Manual all set, all clear,
– IXA 1 777 776 → RØ and 000 001 → RØ
– Execute IXA # 1 Tag Routine
– Check for IXA ⇒ ZØLOWER: 000 001 and 777 776
– Execute IXA # 1 Tag Routine.
RØ → XØ (if J = 16 + 17)
DES Load at 5.3
XØ
– Check Manual all set, all clear, XØ, AØ etc.
ARITHMETIC
– Check if XØ → AØ ± Zero
AØ
– Check Sign extend 18–35 if J 17 and 2^17 = 1
ZØ u
ZØ 1
updated Xm if h = 1 and X ≠ Ø (Xfer at 4.1)
Ueff if J = 16+17 (Xfer at 3.1 T1)

2.4 R1 & R2 CHECKOUT PROCEDURE

- Static checkout:
  Set & Clear via Maint. Panel:
  $F\emptyset_u$, W1, W2, W3, BI, BD, R1 & R2
- MA CLR
- all disables
- Load ICR Index Reg.: $X_i = +1$, $Xm = 0$
- F∅ = 10 00 00 01 2 0———0
- Run — Observe R1 & R2, all bits should be present (incrementing)
- add F∅u — Bits 1 by 1, observing R1, R2
- Restore F∅
- add BI and BD — Bits 1 by 1 (Bits ∅ — 8) observing R1 & R2.
- Execute Memory Tag Routine 5.6
- Execute ICR Tag Routine 5.5

2.5 **R1 or R2 → RSEL → W4 → P**

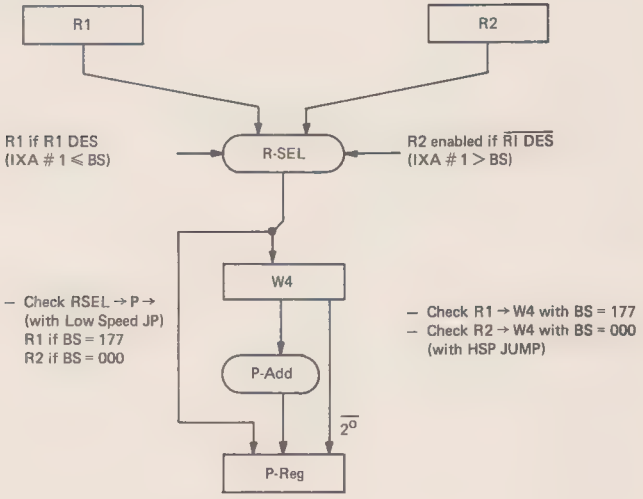
High and Low Speed Jumps, R-Sel

	f	j	a	x	hi	u	
– Setup FØ:	74	04	00	01	0	00000	(f.HSP JUMPS)
	74	15	00	01	0	00000	(f.LOW SP JP)

– all Disables except Advance P, ARITH

– BS = 177 to Check R1

– BS = 000 to Check R2



Checkout Procedure:

– Start with ICR CLR

– Continue by adding Bits → ICR (with ICR clear)
Note: Do not press ICR CLR while CPU is running

– When reaching 777 777$_8$, Result will be +Ø
because of the Index Adders

– Check BD-DES (if set every Jump according to whether BI or BD address was used:
if Jump with R1, BD reset
if Jump with R2, BD set

– Execute Core Memory Tag Routine [5.6]

– Execute P-Checkout [5.8]

2.6 MAIN CONTROL

LIST OF ALL THE DESIGNATOR FLIP FLOPS, IN SEQUENCE OF THEIR PAGE NUMBER

ICR RD U (p. 1)	This FF is set when the operand is read from ICR rather than Core Memory (Ue/f < 200 & Read).
ICR WR U (p. 1)	This FF is set when the operand is written (stored) into ICR rather than Core Memory (Ue/f < 200 & Write).
CORE RD U (p. 2)	This FF sets when the operand was requested from Core Memory. It will control the gating of CORE SEL NO 1 → XØ or X1.
RØ → XØ DES (p. 2)	This FF sets when a Read Op. and J=16+17 (and FØ < 70). It will control the gating of RØ → XØ (immediate operand).
NI DES (p. 2)	This FF sets when the next instruction has been requested from Core Memory. It will control: – CORE SEL NO 2 → FØ – start the T3-chain – enable BRKPT stop if selected – enable setting of NI'DES.
NI'DES (p. 2)	This FF will always set after the NI DES. It will control: – enable Abort sequence if SKIP – enable Illegal Instr. Fault – enable T3 for fault or skip.
IFR DES (p. 2)	This FF sets during the execution of a Load PSR Instn. It will control the loading of the PSR (p. 26).
73 J=17 FF (p. 2)	This FF sets during the execution of a Test & Set Instruction and will control the proper operation thereof.
PASS 2 (p. 3)	This FF sets during Double Precision operations and will control the proper operation thereof.
PASS 2' (p. 3)	This FF sets with PASS 2 and will cause the 2nd Phase of the Double Precision Operation.
INDIRECT (p. 3)	This FF sets when indirect addressing is specified. It will cause the lower 22 bits of the function word to be exchanged by the bits requested from Core Memory (as operand).
EI DISABLE (p. 3)	This FF sets when the 75–15 Instruction is executed. It will prevent all EI (External Interrupts) from being accepted. It is also set by MA CLR.

ABORT (p. 4)	This FF sets during Jump Instructions (Arith. Jump) when the condition is met. It is also used on the 70 JGD-Instruction. Skip can also set Abort. Setting of this FF will cause the present function (=NI) to be aborted so that either the Jump address or NI+1 is next.
ABORT' (p. 4)	This FF controls the second part of the Abort sequence and is set by SKIP + FAULT + ABORT.
BD DES (p. 4)	This FF sets or clears whenever a Jump is taken. It will memorize whether the Jump address was modified with the BD or the BI-portion. It is used during the Instructions SLJ & LMJ to subtract the correct constant from P (influences the B-Selector).
RI DES (p. 4)	This FF can set every 2.3 TØ. It will set if IXA No 1 ≤ BSR (=R1 → S). It controls the R-SEL gating which is used during Jump Instructions.
MSR 00, 01, 02 (p. 4)	The MSR FF's can be set: – from the Maintenance Panel – from the Operators Console – from the ACU – by SIL (73 15) Instruction. It will relocate all Hardware assigned addresses.
W4=ZERO (p. 5)	This FF is used during repeated Instructions (Block Transfer, Search etc.) and will set when the repeat count has been decremented to zero. It will initiate termination sequence (T2).
RPT SETUP (p. 5)	This FF will set with repeated Instructions and causes the contents of the Repeat counter (101) to be loaded into W4.
RPT IN PROGRESS (p. 5)	This FF will set after the Repeat setup sequence and remains set until termination occurs. It will prevent the Next Instruction from being requested.
LAST ADDRESS 15–17 (p. 5)	These FF's can be set by the 73–16 Instruction. They will influence the Memory 1 Parity Error Interrupt Location.
ARITH JUMP DES (p. 6)	This FF sets with conditional Jump Instructions. It enables the Jump sequence if the proper condition is fulfilled.
HIGH SPEED JUMP' DES (p. 6)	This FF is set by the High Speed Jump designator FF and controls the R SELECTOR and P-control and update.
HIGH SPEED JUMP DES (p. 6)	This FF sets with all unconditional Jumps (JIC, JOC, JFC, JUMP, PAIJ, AAIJ, LMJ, HJ). It will produce R1+R2 → S (even if Ue/f < 200).
INTERRUPT (p. 7)	This FF sets with an Interrupt request from the I/O control. It will cause the Interrupt sequence to take place.

INTRPT' (p. 7)	This FF sets with the INTRPT FF and will control the second part of the Interrupt sequence.
PRGM INTRPT DIS (p. 7)	This FF sets with the 72.13 (PAIJ) Instruction. It will prevent the INTRPT FF from setting when a INTRPT REQ is present.
IPI NO 1 (p. 8)	This FF sets with the 73.14 (III) Instruction. It will produce the IPI 1 line to the other 2 processors.
IPI NO 2 (p. 8)	This FF sets with the 73.14 (III) Instruction. It will produce the IPI 2 line to the other 2 processors. (?)
D7, 6, 5, 4 D3, 2, QW, 8 (p. 8)	These FF's set with the 72.15 (LPS) Instruction. For detailed description see Processor State Register and Quarter Word.
TIME OUT ONE SHOT & I/O REQ ONE SHOT (p. 9)	These 2 FF's make sure that no I/O request can be locked out for more than 100μs when operating in User mode. If this happens, a Time Out Fault Interrupt will occur. When the I/O Request O.S. is set, time out fault is disabled (by Guard Mode or Intrpt Req).
A-CONFLICT (p. 10)	This FF will set when an attempt is made to read and write the same ICR Register during cycle 3 TØ. It will prevent reading. It is not operational for the 26, 46 & 74.13 Instructions.
B-CONFLICT (p. 11)	This FF sets when a B-conflict (X-conflict) situation is detected. It will stop and delay the TØ chain (375 ns).
SEL JUMP 1—15 (p. 14)	These FF's can be set either from the Operators Console or the Maintenance Panel. Their condition can be tested with the 74.04 "Jump on Keys" Instruction.
GUARD MODE FAULT (p. 16)	This FF sets when a Guard Mode fault condition is detected. It will set the FAULT DES (which causes a Fault Interrupt sequence to Loc. 243_8).
EXEC RETURN (p. 15)	This FF can be set with the 72.11(ER) Instruction. It will cause a Fault Interrupt to Location 242_8.
ILLEGAL INST FAULT (p. 16)	This FF sets when an illegal Instruction is decoded in the FØ-Register. It will cause a Fault Interrupt to Location 241_8.
MEM 4 FAULT MEM 3 FAULT MEM 2 FAULT MEM 1 FAULT	These FF's set when a Memory Parity Error is detected in the appropriate Module. They must be enabled by the TØ Last FF (which indicates that the Fault occurred during TØ). The Interrupt Locations are the following: MEM 4 237_8 MEM 3 236_8 MEM 2 235_8 MEM 1 LAR-1.

TØ LAST DES (p. 16)	This FF sets every 4.3 TØ and remains set until an I/O sequence is started. It enables the Memory Fault Interrupts (because faults during I/O are handled differently).
ICR PARITY ICR FAULT (p. 16)	These FF's set if a parity error is detected during TØ cycles. It will cause a Parity Error Fault Interrupt to Location 240_8.
FAULT DES (p. 17)	This FF can be set by the following: Exec Return, Illegal Inst Fault, Guard Mode Fault, Mem Fault, ICR Fault, Arithmetic Fault, Test & Set. It will cause a Fault Interrupt sequence to the appropriate location.
TEST AND SET (p. 17)	This FF sets during the Test and Set Instruction if Bit 30 of the tested location is found on. It will cause an Interrupt to Location 244_8.
$F1_{22}$ (CNTL) (p. 26)	This FF sets with Bit 22 of FØ (=A01). It enables Clear and Load of the LAST ADDRESS Register during the 73.16 (LLA) Instruction.
R1 WAIT ODD (p. 28C)	This FF sets when R1 has been requested from an odd Module. It will cause the TØ Operand sequence to synchronize with the Memory.
R1 WAIT EVEN (p. 28C)	This FF sets when R1 has been requested from an even Module.
R2 WAIT ODD (p. 28D)	This FF sets when R2 has been requested from an odd Module.
R2 WAIT EVEN (p. 28D)	This FF sets when R2 has been requested from an even Module.
P WAIT ODD (p. 28E)	This FF sets when P (NI) has been requested from an odd Module. It will cause the TØ Instruction sequence to synchronize with the Memory.
P WAIT EVEN (p. 28E)	This FF sets when P has been requested from an even Module.
IAR_{R18} → MEM 2,3,4 (p. 30)	This FF is used during I/O operations to write the Inp. Amp (shifted right 18 pos.) into Memory 2, 3 & 4.
IAR_{R18} → MEM 1 (p. 30)	This FF is used during I/O operations to write the Inp. Amp. (shifted right 18 pos.) into Memory 1.
IXAI, W5 → MEM 2,3,4 → MEM 1 (p.30)	These FF's are used to transfer the content of IXA No 1 and W5 (=ESI ACW) into Core Write Selector of the selected Memory.

IAR → MEM 2,3,4 (QTR) IAR → MEM 1 (QTR) (p. 30)	These FF's are used to transfer the content of the IAR into Core Write Selector of the selected Module, ESI QTR Word control.
IAR → MEM 2,3,4 → MEM 1 (p. 30)	These FF's are used during ISI input, EI and certain ESI-QTR word operations to transfer the content of IAR to the Core Write Selector of the selected Module.
ARITH COMP → MEM (p. 30)	This FF is used with the 02 & 03 Instructions to store the shifter output → memory (complemented).
ARITH → MEM (p. 30)	This FF is used for all the other Store Instructions to transfer the shifter output → memory.
QTR WD WRITE (p. 32G)	This FF is used to indicate QTR Word operation to the selected Module during a Store or Block Transfer Instruction if the QTR Word designator of the PSR is set.
M NO SEL 2 EVEN/ODD (p. 33A, B)	One of these FF's set will enable the Data from Core Memory to be loaded into FØ, V7, V8.
M NO SEL 1 EVEN/ODD (p. 33A, B)	One of these FF's set will enable the Data from Core Memory to be loaded into RØ & WS, XØ, X1.
STORE DES (p. 34)	This FF sets with all Instructions requiring T1 chain to Store Result ICR (single & double precision). It will initiate T1 chain at 1.2 TØ time.
3T WRITE (p. 34)	This FF is set for all Instructions except 26, 46, 74, 13 where the A-Conflict does not work, and the result is written during cycle 2 T1 already. It prevents writing during cycle 3 T1 when reset.
ICR-WR UPPER (p. 37)	This FF sets when writing of the upper half word of ICR is required. It will enable the "Upper Data timing" which gates $ZØ_U$ into ICR.
ICR-WR LOWER (p. 37)	This FF sets when writing of the lower half word of ICR is required. It will enable the "Lower Data timing" which gates $ZØ_L$ into ICR.
TØ LOCK OUT (p. 43) (=IN LOCK OUT)	Sets at 1.1 04 TØ and prevents the TØ from starting over again before it is reset. It can also be set by power loss (when the voltage goes below a certain limit).
OP LOCK OUT (p. 43)	Sets with 1.1 TØ & Ø4 and prevents the TØ from starting over again before it is reset. It resets with 5.1 OP 04 in normal cycles or 6.1 OP Ø4 if the PARTIAL WRITE FF is set.

PARTIAL WRITE (p. 45)	This FF sets with Store or Block Transfer and a j-Factor between 5–15 or if QTR Word mode. It will cause an additional delay cycle between TØ cycle 5 and the next cycle 1 of TØ. Also set by Test & Set Instruction (73.17).
INST RD EN (p. 46)	This FF sets with various possibilities. It will reverse Operand and Instruction Sequence (when R1+R2 was requested but should be loaded → FØ).
DISABLE RELATIVE INDEX (p. 63B)	This FF sets when Bit 16 of FØ (= i-bit) is on. It enables the disable of BD & BI if in Exec Mode.(Disables Relative Address Modification).

EUROPEAN EDUCATION AND TRAINING CENTER

SECTION 3

ICR

CONTENTS LIST

		Page No.
3.1	TROUBLE SHOOTING HINTS FOR ICR	1
3.2	ICR CHECKOUT PROCEDURES (Write Drivers)	2
3.3	ICR CHECKOUT PROCEDURES (Read Drivers)	3
3.4	ICR CHECKOUT PROCEDURES (Base Switches)	4
3.5	ICR CHECKOUT PROCEDURES (ZØu & Write Drivers)	5
3.6	ICR CHECKOUT PROCEDURES (Sense Amplifiers)	7

TABLES

3.1	WRITE DRIVER INFORMATION	3
3.2	READ DRIVER INFORMATION	4
3.3	WRITE DRIVER BASE SWITCH INFORMATION	4
3.4	READ DRIVER BASE SWITCH INFORMATION	5
3.5	DATA DRIVER INFORMATION	6
3.6	SENSE AMPLIFIER OUTPUT INFORMATION	7

ILLUSTRATIONS

3.1	WRITE DRIVER OUTPUT	2
3.2	READ DRIVER OUTPUT	3
3.3	WRITE DRIVER BASE SWITCH OUTPUT	4
3.4	READ DRIVER BASE SWITCH OUTPUT	5
3.5	DATA DRIVER OUTPUT	6
3.6	SENSE AMPLIFIER OUTPUT WAVEFORM (ZEROS)	7

3 ICR

3.1 TROUBLE SHOOTING HINTS FOR ICR

It is useful to stop the CPU quickly when an ICR fault occurs. To do this jumper 3D04—20 (Clock Stop) to 4B73—09 (ICR Parity).

This will determine in which location the error is occurring and might be possible to determine which bit is dropping, if any (it might be necessary to remove the jumper in order to start the CPU).

If the ICR fault is solid:

1) Write each location to all bits, then zeros and instruction step through ICR to check if there is a pattern (e.g. Base Switches control 20 locations each) in the errors, or bits dropping.

2) Write ones through ICR, in another location write a pattern, check if the pattern reads back correctly. Now write zeros through ICR, in another location write a pattern, if all zeros are read back, suspect a dual read select. Check using patterns that zeros get precedence over ones on read back.

   It is probably quicker to put the Read Address card on an extender and check that no selects are permanently low than to logically troubleshoot further.

   P.S. Expect faults if ICR is run with a card on extender.

3) Using a 06 instruction (06j = 0 a = 1 b = 1 h = 1 u = 0) (earth not desirable)
   IXR 1 = 0 - - - 01/0 - - - 02

   Address tag ICR, then read back through ICR and check tagging. If the tagging is for instance:

   2, 3, 4, 5, 6, 7, 9, 9, 10, 11

   then suspect a dual write select between locations 8 and 9.

   Again the best approach is to put the Write Address Select card on an extender.

4) If no read select is generated, all bits will be gated out of ICR.

5) Check with a scope that all ICR signals are as they should be i.e. Base Switches, timing signals, read sense outputs, etc.

6) Intermittent faults may require data to be correlated over a period, but give the ICR a good margin test first.

7) Remember that if the data is correct the parity bits or parity tree might be faulty.

ICR Timing Card Taps

1108	1106
E1 – E12	E1 – E13
E2 – E14, 13 or 12 (15 With MMA 739s)	E2 – E18
E3 – E22	E3 – E22
E4 – E27	E4 – E27
E5 – E14	E5 – E14
E6 – E23 (24, 25)	E6 – E25
E7 – E14 (15)	E7 – E20
E8 – E20	E8 – E24
E9 – E10 or 11	E9 – E19

The above taps are given as a guide, should a wire "fall off". In practice they are likely to vary slightly from this.

3.2 ICR-CHECKOUT PROCEDURES

To checkout ICR Write Drivers follow the described procedure.

```
                f  j  a  b  h  l        u
1) FØ reg = 05  0  0  0  0  0  0  (ICR Address)
```

Disable Adv P, Real Time Clock, Dayclock, Arith. Interrupt, Memory Fault Interrupt, Read Next Inst., and Arith. Seq. Leave Rate at Normal, insert into AØ reg. any data. Hit Run switch.

2) Check Write Drivers for proper output signal. Refer to Table 3.1 for address and test point information and Figure 3.1 for correct waveshape.

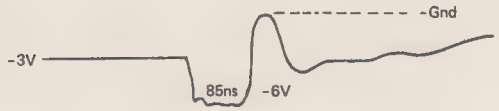


Figure 3.1: Write Driver Output

ADDRESS	CARD	TP
000	J42A	11
001	J42A	15
002	J40A	11
003	J40A	15
004	J38A	11
005	J38A	15
006	J36A	11
007	J36A	15
010	J29A	11
011	J29A	15
012	J27A	11
013	J27A	15
014	J25A	11
015	J25A	15
016	J23A	11
017	J23A	15

Table 3.1: Write Driver Information

3.3 ICR-CHECKOUT PROCEDURES

To check Read Drivers, follow procedure

f j a b l h u

1) FØ reg = 10 0 0 0 0 0 (ICR Address)

   Disable Adv. P, Real Time Clock, Dayclock, Arithmetic Interrupt, Memory Fault Interrupt, and Read Next Inst. Leave Rate control at Normal. Hit Run switch.

2) Check Read Drivers for the proper output. Refer to Figure 3.2 for the proper waveform and Table 3.2 for address and test point information.

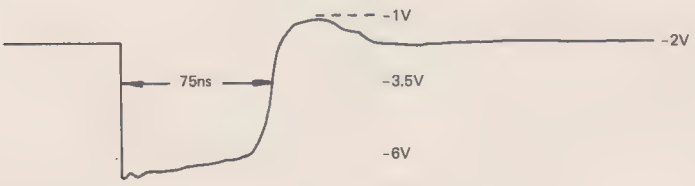


Figure 3.2: Read Driver Output

ADDRESS	CARD	TP	ADDRESS	CARD	TP
000	J41A	10	010	J30A	10
001	J41A	12	011	J30A	12
002	J39A	10	012	J28A	10
003	J39A	12	013	J28A	12
004	J37A	10	014	J26A	10
005	J37A	12	015	J26A	12
006	J35A	10	016	J24A	10
007	J35A	12	017	J24A	12

Table 3.2: Read Driver Information

3.4 ICR-CHECKOUT PROCEDURES

To check Read and Write Base Switches

1) Check the Write Driver Base Switches for proper output. Refer to Figure 3.3 for the correct waveshape and Table 3.3 for address and test point information.

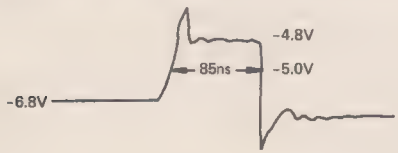


Figure 3.3: Write Driver Base Switch Output

ADDRESS	CARD	TP
000	J42A	13
020	J40A	13
040	J38A	13
060	J36A	13
100	J29A	13
120	J27A	13
140	J25A	13
160	J23A	13

Table 3.3: Write Driver Base Switch Information

2) Check the Read Driver Base Switches for proper outputs. Refer to Figure 3.4 for proper output waveform and Table 3.4 for address and test point information.

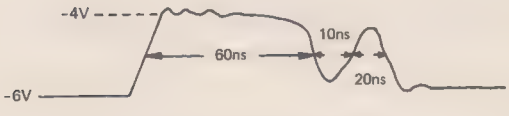


Figure 3.4: Read Driver Base Switch Output

ADDRESS	CARD	TP
000	J41A	14
020	J39A	14
040	J37A	14
060	J35A	14
100	J30A	14
120	J28A	14
140	J26A	14
160	J24A	14

Table 3.4: Read Driver Base Switch Information

IMPORTANT NOTE: The following intercommunication was dispatched from Roseville Technical Operations. Memo reads:

> "It has been found that pulling I.C.R. Read Driver Board 4033049–01 with power up, will cause a failure in the board pulled plus failures in various other P.C. Boards in a random pattern.
>
> The cause is from a component being furnished by an alternate vendor. The component problem is in the process of being resolved. In the meantime, it is suggested power be down on the 1108 before pulling any P.C. Board. This should apply to all P.C. Boards and not only to those which are part of the ICR."

3.5 ICR-CHECKOUT PROCEDURES

ZØu & L Write Drivers

f j a b h l u
1) FØ = 05 0 0 0 0 0 (any ICR address)

- Disable Adv. P, Real Time Clock, Dayclock, Arithmetic Interrupt, Arith. Seq., Memory Fault Interrupt, and Read Next Instruction.
- Rate control Normal.
- Insert Data into AØ register.
- Depress Run switch.

2) With zeros in AØ, check the outputs of the Data Drivers ("0"TP) for the proper signal. Refer to Figure 3.5 for the proper waveform and Table 3.5 for data and test point information.

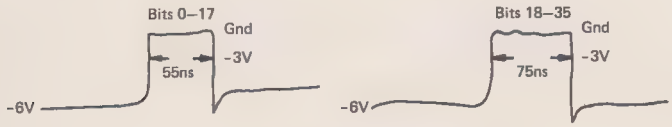


Figure 3.5: Data Driver Output

BIT	LOC	"0"TP	"1"TP	BIT	LOC	"0"TP	"1"TP
0	J44B	2	4	18	J33B	2	4
1	44B	5	11	19	33B	5	11
2	44B	12	13	20	33B	12	13
3	J43B	2	4	21	J30B	2	4
4	43B	5	11	22	30B	5	11
5	43B	12	13	23	30B	12	13
6	J41B	2	4	24	J28B	2	4
7	41B	5	11	25	28B	5	11
8	41B	12	13	26	28B	12	13
9	J39B	2	4	27	J26B	2	4
10	39B	5	11	28	26B	5	11
11	39B	12	13	29	26B	12	13
12	J37B	2	4	30	J24B	2	4
13	37B	5	11	31	24B	5	11
14	37B	12	13	32	24B	12	13
15	J35B	2	4	33	J22B	2	4
16	35B	5	11	34	22B	5	11
17	35B	12	13	35	22B	12	13

Table 3.5: Data Driver Information

3.6 ICR-CHECKOUT PROCEDURES

Sense Amplifiers

1) With the computer running, check the output of the Sense Amplifiers. Refer to Figure 3.6 for the correct waveform and Table 3.6 for the card location and test point information.

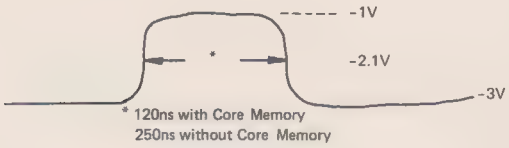


Figure 3.6: Sense Amplifier Output Waveform (Zeros)

2) Write ICR addresses 0, 14, 15, 20 to ones and set up a Read ICR Address 20 Instruction. Output of all sense amps should be a constant -3 volts. Refer to Table 3.6 for output information.

BIT	LOC	TP	BIT	LOC	TP	BIT	LOC	TP
0	J42B	2	12	J36B	2	24	J29B	2
1	42B	10	13	36B	10	25	29B	10
2	42B	13	14	36B	13	26	29B	13
3	42B	15	15	36B	15	27	29B	15
4	J40B	2	16	J34B	2	28	J27B	2
5	40B	10	17	34B	10	29	27B	10
6	40B	13	18	34B	13	30	27B	13
7	40B	15	19	34B	15	31	27B	15
8	J38B	2	20	J32B	2	32	J25B	2
9	38B	10	21	32B	10	33	25B	10
10	38B	13	22	32B	13	34	25B	13
11	38B	15	23	32B	15	35	25B	15

Table 3.6: Sense Amplifier Output Information

EUROPEAN EDUCATION AND TRAINING CENTER

SECTION 4

INPUT/OUTPUT

CONTENTS LIST

		Page No.
4.1	HOW TO OUTPUT VIA I/O CHANNEL	1
4.2	HOW TO INPUT FROM I/O CHANNEL	1
4.3	FORCE BOOT 5008	1
4.3.1	Force Boot from Any Device	2
4.4	FORCE BOOT VIA MSA	2
4.5	I/O CONTROL LOGIC	2

4. INPUT/OUTPUT

4.1 HOW TO OUTPUT VIA I/O CHANNEL

- EF (Function –V7/V8) *
- CH # → Acknowledge Register
- Set ACK EF 0–7 or 8–15 (outputs Function)
  (CU waits with ODR now but Channel not active yet)
- Load ACW (by means of ICR clear)
- Set Out Active for this channel
  (Tape should start moving when first OA received)

4.2 HOW TO INPUT FROM I/O CHANNEL

- Function → V7/V8 *
- CH # → ACK Register
- Load IACW (with ICR clear) desired Block length → W
- Set IN ACT
- Set ACK EF 0–7 or 8–15

4.3 FORCE BOOT 5008

CPU:
- MA CLR
- Do not Start
- Set AØ ARITH 2^{31}
- ICR CLR (writes IACW: G=00_2 W=020000_8 V=0 0)
- Set IN ACT on BOOT-Channel
- Subsystem CLR on Console

5008:
- Boot Channel selected (can be any)
- 7/9 channel correct
- Test Mode, MA CLR
- Enter Function: 4_8
- Enter Servo # → Unit Select
- Run
- Repeat as follows:
- MA CLR
- Func 4_8 – Func Reg.
- Servo # → Unit Select
- Run, immediately switch to On-Line (Test off) while Tape is still rewinding

CPU:
- Should have loaded $20'000_8$ words (which can be seen at ZØ:
  G=00, W=00–00, V=$020'000_8$)

* can include DA if MSA! or Unit # if 5008

4.3.1 Force Boot from any Device

CPU: – Select Boot Channel
– press Initial Load switch
– add bits to W part of XØ to load in more than 02000 words
– press Run

Note: CPU hangs if the words are transferred.

4.4 FORCE BOOT VIA MSA:

CPU: – MA CLR
– Set
– Set AØ Arith 2^{31}
– ICR CLR (set up ACW: G W V 00 020000 000000)
– Set IN ACT on MSA-Channel
– Subsystem CLR on Console

MSA: – Boot-Function must be patched correctly (A50)
– Gnd TP8 on A47 (sets BOOT FF)
– Set TEST MODE (OFF-LINE)
– Select proper Device Address
– All other Switches Off!
– Start (Tape should move, reading offline, preceeded by a rewind)
– MA CLR
– Select Device Address
– Start, and immediately switch to On-Line
– Tape should rewind, then read this time On-Line

CPU: – Should have loaded $20'000_8$ which can be seen at ZØ
G W V
00 00000 020000_8

MSA: – Status (RIW-Reg. should be
DA, Status = $0C_H$; E, ABC = Ø)

4.5 I/O CONTROL LOGIC

LIST OF THE CONTROL FLIP FLOPS, IN SEQUENCE OF THEIR PAGE NUMBER

OUT FF (p. 601)	This FF will set with an unblocked ODR on Channel Ø–15. If this FF is set, all other Functions are disabled (p. 603).
IN FF (p. 601)	This FF will set with an unblocked IDR on Channel Ø–15. A Dayclock IDR will also set IN. This FF disables all lower priority Functions but can be disabled by OUT (p. 603).

RTC DEC (p. 601)	This FF will set with an RTC DEC request (RTC one shot gated by OSC). The output of this FF is used to start a priority scan (p. 605, 603).
PWR LOSS + PAR INTRPT (p. 601)	This FF will set: a) if the AC power loss condition is detected, b) if one of 3 different kind of parity errors during T8 is detected. It will enable a priority scan & if highest, an INTRPT will occur.
EI ESI (p. 601)	This FF will set with an unblocked EI request on an ESI channel. It starts the priority scan & controls the proper operation of an external Interrupt ESI sequence.
IN MON ESI	This FF sets during an ESI IDR service if Terminate condition exists. It will start a priority scan and controls the operation of IN MON ESI Interrupt sequence.
OUT MON ESI	This FF sets during an ESI ODR service if Terminate condition exists. It will start a priority scan and controls the operation of OUT MON ESI Interrupt sequence.
RTC INTRPT (p. 602)	This FF sets if an RTC Intrpt condition is detected (EA8 during RTC DEC). It will start a priority scan and controls the operation of RTC Interrupt sequence.
EI ISI	This FF sets with an unblocked EF request from an ISI channel. It will start a priority scan and controls the operation of an ISI EI Interrupt sequence.
IN MON ISI	This FF sets during an ISI IDR service if a Terminate condition exists. It will start a priority scan and controls the operation of IN MON ISI Interrupt sequence.
OUT MON ISI (p. 601)	This FF sets during an ISI ODR service if a Terminate condition exists. It will start a priority scan and controls the operation of OUT MON ISI Interrupt sequence.
FUNC MON ISI	This FF sets during an ISI (Function) ODR service if a Terminate condition exists (no more Function to be output). It will start a priority scan & controls the operation of FUNC MON ISI Interrupt sequence.
SYNC I	This FF sets if the Sync I (Interprocessor Interrupt line is raised). It will start a priority scan and controls the III Interrupt.
SYNC II	This FF sets if the Sync II (Interprocessor Interrupt line is raised). It will start a priority scan and controls the III Interrupt.
0 – 7 INL.O.	This FF can set if Channels Ø–7 are selected in Compatible Mode and Channel Ø–7 is being served (any pri.). It will prevent any IDR, ODR and EI on Channel Ø–7 from being served until reset from the ACK Counter.
0 – 7 OUT L.O. (p. 604)	This FF sets: a) if an OUT-Function is served on Channel Ø–7, b) if any Function is served on Channel Ø–7 and Channel is selected in Compatible Mode.
8 – 15 IN L.O.	This FF sets if Channel 8–15 are selected in Compatible Mode and any Function is being served (IDR/ODR/EI). It will lockout Channels 8–15 until reset.

8 – 15 OUT L.O.	This FF sets: a) if an OUT Function is served on Channel 8–15, b) if any Function is served on Channel 8–15 and Channel is selected in Compatible Mode. It will Lockout Channels 8–15 until reset.
INTRPT REQ (p. 605)	This FF sets with any Interrupt Function getting Function priority. It will make an Interrupt Request (set INTRPT FF if enabled).
INTRPT LOCKOUT (p. 605)	This FF sets with the same conditions as the INTRPT REQ FF. It makes sure that no other I/O Interrupt can be served until reset again (by AAIJ).
FUNC PRI DIS (p. 605)	This FF sets 2 Phases after the first Function requesting service. It disables any further Functions from being added.
FUNC PRI DIS' (p. 605)	This FF sets after the FUNC PRI DIS FF is set. When set it enables CHAN PRI DIS FF to set. (The purpose of FUNC PRI DIS' is mainly to extend the time of the Channel Priority scan to one cycle).
CHAN PRI DIS (p. 605)	This FF sets after the FUNC PRI DIS' FF is set. When set, it disables any further channels from entering the Channel Priority reg.
VALID FP (p. 605)	This FF sets during the Function & Channel Priority scan but only if IDR, ODR, RTC DEC, EI (ISI + ESI) Function has priority. It disables T9 It enables T8 It locksout T0.
DAYCLOCK DIS (p. 605)	This FF can set with the 73–14 Instruction and a=12. It can be reset by MA CLR or 73–14 and a=11. When set, Dayclock IDR and Interrupt are disabled.
ISI (p. 606)	This FF sets every 1.1 T8. However, it resets again if EI (ISI + ESI) or ESI EF or RTC or DAYCLOCK IN. Its main purpose is when reset, to disable reading and updating of the ICR ACW.
DAYCLOCK (p. 606)	This FF sets with DC IDR and DC EC. It will change a normal IDR operation into a DC IDR. A DC EI will also be changed from an ISI EI. If this FF is set, the ACK FF's (p. 613) will not set.
ESI – 1 FF (p. 611)	This FF sets with an ESI IDR or ODR if this channel has priority. It controls the start of any of the 3 ESI cycles (RC,WC,R/W).
ESI – 2FF	This FF sets after the ESI–1 FF. It controls the end of the 3 ESI cycles.
RC – 1 (p. 612)	This FF sets when an ESI IDR or ODR gets priority. It controls reading out the ACW from Core Memory.

RC – 2	This FF sets during the ESI RC (after the RC–1 was set). It controls parity error checking and reading out the ACW from Core Memory. It will remain set until the WC–1 FF is set (making sure this is the next cycle).
WC – 1	This FF is enabled by RC–2 and sets when the ESI operation gets priority again. It will control updating of the ACW and writing back into Core Memory.
WC – 2	This FF sets during the ESI WC (after the WC–1 was set). It will remain set until the next ESI cycle is started, making sure R/W cycle is next). Exception: if WC and R/W overlap, reset is enabled early.
R/W – 1 (p. 612)	This FF can set: a) After WC if Access Control Word and Data are located in same Memory (65K). In this case ALT FF is reset. b) At the same time as WC–1 if Access Control Word and Data are located in different Memories (65K). In this case ALT FF is set. It controls the proper operation of the R/W cycle: if IDR input (write) Data into Memory , if ODR output Data to ODR. It also controls setting of the ACK Control FF's.
R/W – 2 (p. 612)	This FF sets after R/W was set. It controls outputting the Data to the ESI device.
ESI ACT (p. 612)	This FF sets with RC–1 FF and resets again with RW–1 (so it is set during the whole ESI operation). It locks out any other ESI channel from being served.
ESI ALT (p. 612)	This FF sets if the ACW and Data are located in different Memories (65K), resets if in same Memory (65K). It enables overlapping of WC and R/W cycles when set.
I/O FF (p. 612)	This FF sets during the RC of an ESI operation if an ODR is being served. It resets if IDR. This FF will produce subsequent ODR/IDR request for WC and R/W.
QTR WD WR CTRL (p. 612 A)	This FF sets with every IDR being served. It enables the ESI Write Control for QTR Word.
QTR WD ESI (p. 612 A)	This FF sets every time an ESI operation (IDR/ODR) gets priority and this channel is selected in Quarter Word Mode. It is producing the second enable to the Core Write control. It also enables the Core Selector ESI.
IN 7–1 (p. 613)	This FF sets if a Channel Ø–7 got priority for Input or EI (ISI or ESI). Its setting is disabled for Dayclock or Abort. When set it enables the ACK Control & T8 to carry on after 5.4.
IN 7–2 (p. 613)	Setting of this FF is enabled by IN 7–1. During the time it is set, IA will be gated out to the channel which got priority.
OUT 7–1 (p. 613)	This FF sets if any channel from Ø–7 got priority for OUTPUT (ODR). Its setting is disabled for Dayclock or Abort. When set, enables the ACK Control & TB to carry on after 5.4.

OUT 7–2 (p. 613)	Setting of this FF is enabled by OUT 7–1. During the time it is set, OA will be gated out to the channel which got priority.
EF 7–1 (p. 613)	This FF sets if any channel from Ø–7 got priority for EF (after LFC). Its setting is disabled for Dayclock or Abort. When set, it enables the ACK Control & T8 to carry on after 5.4.
EF 7–2 (p. 613)	Setting of this FF is enabled by EF 7–1. During the time it is set, EF will be gated out to the channel which got priority.
IN 8–1 IN 8–2 OUT 8–1 OUT 8–2 EF 8–1 EF 8–2 (p. 613)	See IN 7–1 See IN 7–2 See OUT 7–1 See OUT 7–2 See EF 7–1 See EF 7–2 But Channels 8 through 15
ACK CONTROL (p. 614)	This FF sets if one of the 7–1 or 8–1 FF's is set and this Group is selected for compatible Mode. It will start the ACK Counter and ACK sent out will be 3.5 µsec. long. It can also be set by Initial Load (which will start the Counter but has effect only if the channel selected is in Compatible Mode).
EF OPERATION (p. 614)	This FF sets with an EF operation having priority on any channel. It enables the reset of the EF FF of that channel which got priority, but only if ESI !!! (If the EF signal produced on p. 615, E4284, was used as the reset enable, this would be lost as soon as the EF channel FF resets).
LC Ø–5 (p. 615)	These FF's are the lower part of the ACK Counter which is enabled by Compatible Channel Mode. It is preset to 07 & by the Set Counter signal.
UC Ø–5	The FF's are the upper Part of the ACK Counter, which is enabled by Compatible Channel Mode. Its Outputs are used to determine the start and finishing time of the ACK signals.
F 1 22–25 (p. 616/617)	These four FF's set with the signal FØ→F1 (which is generated during LOC, LIC, & LFC). They will assume the state of CSR or F@a, which specifies one channel from Ø–15.
CSR 00–03 (p. 616/617)	These four FF's (= Channel Select Register) will assume the state of the Channel Select switches during Initial Load. It can also be loaded with the 73–16 (LCR) Instruction. It is cleared by the same Instruction and (U) = Ø. Its prime function is to maintain the Boot Channel Number (a or CSR) until the Software knows which channel is booted from.
DAYCLOCK EI (p. 618)	This FF is set by MA CLR or the absence of a Dayclock EI signal. It will reset if DC EI is granted priority. It makes sure that this request (once honoured) does not enter the priority again until the request is dropped.

DAYCLOCK IDR (p. 618)	This FF is set by MA CLR or the absence of a DC IDR signal. It will reset as soon as DC IDR is granted priority. It makes sure that this request does not enter the priority again until the request is dropped.
SYNC II OS SYNC I OS	These FF's set by the absence of the IPI Ø or 1 signal. It will reset once the request has gotten priority, making sure that this request does not enter the priority again until dropped.
RTC INTRPT (p. 618)	This FF sets if an EAB condition is detected during an RTC DEC operation. It will request service for RTC INTRPT. Its setting is disabled by: DIS RTC (on Maint. Panel), Initial Load and Run.
RTC OS (p. 618)	These 2 FF's make up the Network for the RTC DC request. They are controlled by the RTC oscillator. The RTC OSC can be disabled by DIS RTC (Maint. Panel), Initial Load and Run.
POWER LOSS (p. 618)	These 2 FF's make up the Network for the PWR LOSS INTRPT Request. They are controlled by Power Loss signal (Absence of AC Power). The One-Shot must be enabled by the switch on the Maint. Panel.
I/O SEQUENCE (T8-Sequence) (p. 624/625)	This timing chain is used to control IDR, ODR and EI Request & Transfers. While T8 is running, TØ is locked out. T8 is enabled by the VALID FP FF.
OBR-Sequence (p. 626)	This timing chain is used to control the loading of V7 and V8 Registers.
ACK-Sequence (p. 626)	This timing chain is used to control the Acknowledge timing on normal channels. (It will produce an ACK signal of the duration of 688ns.).
INTRPT-Sequence (T9-Sequence) (p. 672)	This timing chain is used to control the initiation of an Interrupt sequence for all the I/O Interrupt Requests except EI. T9 is enabled by the absence of Valid FP.
I/O ABORT-Sequence (p. 628)	This sequence will start: a) In ISI IDR/ODR if W2 = Ø. b) In ESI or ISI, IDR/ODR/EI if T8 has been started, and no CHAN PR1 (cleared by LOCKOUT). c) If EI ISI got priority but there is a blocked EI ESI. This is to make sure that EI ESI gets higher priority than EI ISI. This sequence will abort T8, and no Data is transferred and the ACW is left unchanged.
INITIAL LOAD-Sequence (p. 628)	This sequence is started with Initial Load (Maint. Panel, ACU or Console) and Run. It controls the proper operation thereof.
EF PAR ERROR (p. 629)	This FF sets if a Parity Error (ICR or Core Memory) occurred during an EF transfer to a Subsystem. It disables the EF signal from being sent out to the S/S.

I/O ICR ERROR (p. 629)	This FF sets if an ICR error occurred during an ISI operation. It will reset IN/OUT ACT for the channel being served. It also disables writing back the updated ACW as well as requesting Core Memory (R2 → S blocked).
ES1 CORE ERROR (p. 629)	This FF sets if a Core Memory occurred during an ES1 operation. It disables R1 → S and R2 → S for this ES1 operation.
W5 = Ø INIT (p. 629)	This FF sets if W5 is detected to be = Ø initially (no Data to be transferred). It disables R1 → S and R2 → S for this ES1 operation.
V4 – REG (p. 629)	These 5 FF's are used to setup the address of the ACW during I/O operation. (To read the ACW the actual FF input is used, = I/O ICR, V4 is only used to write back the updated ACW, = V4).
V5 – REG (p. 630)	These 5 FF's are used to setup part of the Interrupt address (for E1: 200, 201, 202, then Intrpt Location, if EI only INTRPT Loc.). They can be set with T8 & EI or with T9.
V6 – REG (p. 630)	These 4 FF's are set equal to the channel which caused the Interrupt. Its content can be obtained by the 72–14 (SCN) Instruction.
POWER LOSS' (p. 707)	This FF sets with the Power Loss OS being set and the Set Func PR1 signal. It disables I/O parity Errors (which might occur, but have lower priority).
PARITY INTRPT REQUEST (p. 707)	This FF sets with any parity error occurring during T8. It sets the PAR ERROR or PWR LOSS INTRPT FF (p. 601) and disables subsequent setting of the I/O Parity Reg. (Which contains the channel which was served when the Par. error occurred).
PAR INTRPT (p. 707)	This FF will set if the Function PWR LOSS + PARITY INTRPT is requested and POWER LOSS'. It will enable I/O Parity Register to be gated → B' Sel, instead of V6. (In order to capture the channel where the error occurred when executing SCN).
I/O PARITY – REG (p. 707)	These FF's set every T8 according to the channel # in V4 Reg. (This is a precaution in case of a parity error). As soon as a Par Error Request exists, subsequent setting is disabled. I/O Parity content can be obtained by the 72–14 (SCN) Instruction.
I/O CW PARITY (p. 707)	This FF sets if a Core Memory parity error is detected during the RC of an ES1 operation. It will gate the Interrupt address (21)1 if not PWR LOSS.
I/O DATA PARITY (p. 707)	This FF sets if a Core Memory Par. error is detected during ISI or ES1/RW. It will gate the Interrupt address (21)3 if not PWR LOSS.
I/O ICR PARITY (p. 707)	This FF sets if an ICR Parity error is detected during a T8 cycle. It will gate the Interrupt address (21)2 if not PWR LOSS.

EUROPEAN EDUCATION AND TRAINING CENTER

SECTION 5

TEST PROGRAMS AND LOOPS

CONTENTS LIST

		List No.
5.1	INTRODUCTION	12
5.2	ERROR DISPLAYS	39
5.3	FIXED ADDRESS ASSIGNMENT	55
5.4	ICR TAG TEST	68
5.5	INDEX ADDER 1 CHECK OUT	98
5.6	J FACTOR TEST – TE, TZ, TNE, ADD	119
5.7	CORE TAG TEST	153
5.8	REPEAT INSTRUCTION USED TO CLEAR AND CHECK MEMORY	181
5.9	CORE MEMORY TEST	202
5.10	ANOTHER MEMORY TEST	231
5.11	P. REGISTER CHECK OUT	277
5.12	TRICK PROGRAM	309
5.13	I/O BACK TO BACK TESTING IN ISI MODE	335
5.14	I/O CABLE WRITING	385
5.15	SHORT I/O BACK TO BACK TEST	416
5.16	LONG INPUT/OUTPUT BACK TO BACK TEST	461
5.17	ESI BACK TO BACK INPUT TEST	522
5.18	ESI OUTPUT BACK TO BACK TEST	643
5.19	CONSOLE 4009 TEST	744
5.20	CONSOLE 4004 TEST	759
5.21	HOW TO WRITE A PROGRAM IN OCTAL INST. CODE	768
5.22	CTMC BACK TO BACK TEST	790
5.23	MSA TURNAROUND TEST	853
5.24	ROUTINE TO BUILD PSR AFTER BOOT	923
5.25	ROUTINE TO SELECT ONE OUT OF 15 TESTS	948
5.26	ROUTINE TO IGNORE INTERRUPTS	966

```
@RUN      GM,R784PC,CETEST                                        ***

@ASG,TJ   TAPE,16N,SAVE                                           ***

@ASM,IS   A,B                                                     ***
ASM 14R1 RL72R1 09/12/76 16:13:54 (,0)
```

```
 1.
 2.
 3.
 4.
 5.
 6.
 7.
 8.
 9.
10.
11.
12.
13.
14.
15.
16.
17.
18.
19.
20.
21.
22.
23.
24.
25.
26.
27.
28.
29.
30.
31.
32.
33.
34.
35.
36.
37.
38.
39.
40.
41.
42.
43.
```

```
.
.
.
.
.
.
.     ****************************
.     *                          *
.     *  TEST PROGRAMS AND LOOPS *
.     *                          *
.     ****************************
. FOLLOWING IS A PROCEDURE, WHICH WILL ALLOW AN ENGINEER TO WRITE LITTLE
. PROGRAMS AND TEST LOOPS AND PUT THEM ONTO A BOOTABLE TAPE.
.
. THE TAPE WILL BE GENERATED UNDER EXEC 8 CONTROL USING THE ASSEMBLER AND
. STANDARD M A P PROCESSOR. THE PROGRAMS CAN BE WRITTEN IN ASSEMBLER LANGUAGE
. OR IN PSEUDO OCTAL CODE (INSTRUCTION FORMAT) IF THE FORM STATEMENT FOR THE
. INSTRUCTION FORMAT IS INCLUDED. THE ADDRESS RANGE FOR ALL PROGRAMS IS 000
. TO 01777. SPECIFIC START ADDRESSES CAN BE FORCED FOR A PROGRAM BY THE
. R E S-ERVE STATEMENT AND THE SEQUENCE OF PROGRAM CARDS.
.
. THE HIDDEN MEMORY (008-177) CAN ONLY BE USED FOR PROGRAMS, WHICH DO NOT
. USE OPERANDS OUT OF THE HIDDEN MEMORY.
. THE INSTRUCTION SEQUENCE FOR THE HIDDEN MEMORY IS WRITTEN IN A DIFFERENT
. I-BANK ($(3)). SPECIAL CARE MUST BE TAKEN TO GET THE PROPER OPERAND ADDRESS
. IN THIS PART OF THE PROGRAM. THEREFORE ALL LABELS REFERING TO $(3) MUST BE
. DECREMENTED BY THE STARTING ADDRESS OF THIS I-BANK (LABEL IS 'B').
.
. THE MEMORY ADDRESSES 0200-0247 ARE INTERRUPT ADDRESSES AND SHOULD NOT BE USED
. FOR PROGRAMS.
.
. N O T E : AFTER BOOTING THE TAPE, AN INPUT MONITOR INTERRUPT TO ADDRESS 022D
. ******* WILL OCCUR. AN EXTERNAL INTERRUPT FROM THE TAPE UNIT IS PENDING
. AND WILL OCCUR AFTER AN A A C I AND A A I J  INSTRUCTION IS EXECUTED.
.
. THE SEQUENCE OF CARDS TO PRODUCE A BOOTABLE TAPE ARE MARKED WITH *** ON THE
. FAR RIGHT HAND SIDE. THESE CARDS MUST BE USED IN THE SHOWN SEQUENCE
. TO GENERATE THE TAPE.
. ERROR DISPLAYS
. **************
. THE ERROR DISPLAY WORKS ON THE BASIS, THAT A HALT INSTRUCTION
. READS THE A-REGISTER SPECIFIED BY THE A-FIELD OF THE HALT INSTRUCTION,
. TRANSFERS THE CONTENTS INTO THE ARITHMETIC XO REGISTER AND STOPS.
```

```
2   44.
    45.                                         .
    46.                                         .
    47.                                         .
    48.                                         .
    49.                                         AXR$                    . SPECIFIES THE A X AND R REGISTER      ***
    50.                                                                 . FOR THE ASSEMBLER
    51.                                         $(1)                    . DEFINES THE FIRST I-BANK              ***
    52.     01   000000                                                 . RESERVES SPACE FOR THE HIDDEN MEMORY ***
                                                RES       0200
    53.                                         A                       . START LABEL OF THE PROGRAM AREA       ***
    54.                                                                 .
    55.                                                                 . FIXED ADDRESS ASSIGNMENT
    56.                                                                 . ************************
    57.             000200  000000000000      STATUS    +         0     . STATUS LOCATION
    58.             000201  000000000000      STATUS1   +         0     . STATUS LOCATION CPU 1
    59.             000202                              RES       0206-$  .
    60.             000206  000000000000      BCHAN     +         0     . BOOT CHANNEL
    61.             000207  000000000000      BPSR      +         0     . P S R STORAGE
    62.                                                 DO  8 , LMJ  X1,IGNORE-B    . PAD 7 LOC. WITH LMJ TO IGNORE INTERRUPTS
                    000210  74 13 01 00 0 000041
                    000211  74 13 01 00 0 000041
                    000212  74 13 01 00 0 000041
                    000213  74 13 01 00 0 000041
                    000214  74 13 01 00 0 000041
                    000215  74 13 01 00 0 000041
                    000216  74 13 01 00 0 000041
                    000217  74 13 01 00 0 000041
    63.                                                                 . -B MUST BE SUBTRACTED TO REFER TO
    64.                                                                 . THE HIDDEN MEMORY.
    65.             000220  74 06 00 00 0 000000      NOP               . MONITOR INT. LOC.  GO TO ZERO
    66.                                                 DO  23 , LMJ  X1,IGNORE-B   . PAT ANOTHER 23 LOCATIONS
                    000221  74 13 01 00 0 000041
                    000222  74 13 01 00 0 000041
                    000223  74 13 01 00 0 000041
                    000224  74 13 01 00 0 000041
                    000225  74 13 01 00 0 000041
                    000226  74 13 01 00 0 000041
                    000227  74 13 01 00 0 000041
                    000230  74 13 01 00 0 000041
                    000231  74 13 01 00 0 000041
                    000232  74 13 01 00 0 000041
                    000233  74 13 01 00 0 000041
                    000234  74 13 01 00 0 000041
                    000235  74 13 01 00 0 000041
                    000236  74 13 01 00 0 000041
                    000237  74 13 01 00 0 000041
                    000240  74 13 01 00 0 000041
                    000241  74 13 01 00 0 000041
                    000242  74 13 01 00 0 000041
                    000243  74 13 01 00 0 000041
                    000244  74 13 01 00 0 000041
                    000245  74 13 01 00 0 000041
                    000246  74 13 01 00 0 000041
                    000247  74 13 01 00 0 000041
```

```
67.                                                    . I C R TAG TEST
68.                                                    . ***************
69.                                                    . THIS TEST USES THE INDEX REGISTER 0,1 AND 2 TO TEST THE ADDRESS
70.                                                    . CAPABILITY OF ALL OTHER I C R LOCATIONS.
71.                                                    . STOP 4 INDICATES AN ERROR AND DISPLAYS THE FIRST TIME THE
72.                                                    . EXPECTED ADDRESS AND THE SECOND TIME THE RECEIVED RESULT.
73.                                                    . IN THE X0 REGISTER.
74.                                                    . IF AN ERROR OCCURS REGISTER A8 WILL BE DESTROYED.
75.                                                    . TEST CAN BE LOADED WITH ANY MSR SELECT
76.                                                    . NOTE: REAL TIME CLOCK MUST BE DISABLED FOR THIS TEST.
77.                                                    . STOP 1 = END OF TEST
78.
79.      01   000250  27 16 00 00 000175     TEST1    LX,U      X0,0175          . REPEAT COUNT IS 0175
80.           000251  27 16 01 00 000002              LX,U      X1,2             .
81.           000252  46 16 01 00 000001              LXI,U     X1,1             . X1 IS 1/2
82.           000253  06 00 01 01 2 000000            SX        X1,0,*X1         . STORE TAG
83.           000254  70 000 00 0 000253              JGD       0,$-1            . REPEAT 0175 TIMES
84.           000255  27 16 00 00 000174              LX,U      X0,0174          . REPEAT COUNT IS 0174
85.           000256  26 16 01 00 000002              LXM,U     X1,2             . X1 IS 1/2
86.           000257  27 00 02 01 2 000001  READ      LX        X2,1,*X1         . LOAD TAG TO X2
87.           000260  25 00 02 00 0 000001            ANX       X2,X1            . SUBTR. TAG FROM PATTERN
88.           000261  50 00 00 00 0 000002            TZ        X2               . RESULT SHOULT BE ZERO
89.           000262  74 04 00 00 0 000265            J         $+3              . ERROR. GO TO DISPLAY
90.           000263  70 000 00 0 000257     REP      JGD       X0,READ          . RESULT IS ZERO. REPEAT
91.           000264  74 05 01 00 0 000250            HKJ       1,TEST1          . STOP 1 = END OF TEST
92.           000265  10 00 10 00 0 000001            L         A8,X1            . LOAD A8 TO DISPLAY ADDR.
93.           000266  74 05 10 00 0 000267            HKJ       8,$+1            . STOP 4 = DISPLAY FAILURE ADDRESS
94.           000267  10 00 10 01 0 000000            L         A8,0,X1          . LOAD A8 TO DISPLAY DATA
95.           000270  74 05 10 00 0 000263            HKJ       8,REP            . DISPLAY DATA IN X0
96.           000271                                 RES       0300-$           .
97.                                                  .
98.                                                  . INDEX ADDER 1 CHECK OUT
99.                                                  . ************************
100.                                                 . THE TEST WILL CALCULATE (A0) + 1 IN THE ARITHMETIC AND (A1) + 1
101.                                                 . VIA I X A  1 AND COMPARE THE TWO RESULTS. A STOP 4 WILL INDICATE
102.                                                 . A MISCOMPARE AND DISPLAY(A) IN THE X0 REGISTER. ANOTHER STOP 4
103.                                                 . WILL DISPLAY THE CONTENTS OF A1
104.                                                 . TEST CAN BE LOADED WITH ANY MSR SELECT
105.          000300  10 00 00 00 0 000312  TEST2    LA        A0,ONEZER        . A0 IS 1/0
106.          000301  10 00 01 00 0 000014            LA        A1,A0            . A1 IS 1/0
107.          000302  74 06 00 15 2 000000            NOP       0,0,*A1          . UPDATE A1 VIA IXA 1
108.          000303  72 04 00 00 0 000313            AH        A0,ONE           . UPDATE A0 VIA ARITH
109.          000304  53 00 00 00 0 000015            TNE       A0,A1            . TEST A0 EQUAL A1
110.          000305  74 04 00 00 0 000302            J         TEST2+2          . O.K. REPEAT
111.          000306  10 00 10 00 0 000014            L         A8,A0            . NO. LOAD A8 FOR DISPLAY
112.          000307  74 05 10 00 0 000310            HKJ       8,$+1            . DISPLAY A0 IN X0
113.          000310  10 00 10 00 0 000015            L         A8,A1            . LOAD A8 FOR DISPLAY
114.          000311  74 05 10 00 0 000300            HKJ       8,TEST2          . DISPLAY A1 IN X0
115.          000312  000001 000000           ONEZER   +         1,0              .
116.          000313  000000 000001           ONE      +         0,1              .
117.          000314                                 RES       0320-$           .
```

```
118.
119.                                                    . J FACTOR TEST --TE,TZ,TNE,ADD--
120.                                                    . ******************************
121.                                                    . THIS IS A BASIC TEST FOR J= 16 AND 17
122.                                                    . N O T E : THE DISPLAY WORKS ON THE BASIS, THAT A HALT INSTRUCTION
123.                                                    . *******  READS THE A REGISTER SPECIFIED BY THE A FIELD. THE STOP
124.                                                    . OCCURS WHEN THE CONTENTS IS LOADED INTO THE X0 REGISTER.
125.                                                    . IN THIS TEST ANY STOP # IS AN ERROR
126.                                                    . STOP 1 WILL REQUIRE MANUAL CMPARISON
127.                                                    . TEST CAN BE LOADED WITH ANY MSR SELECT
128.    01   000320  10 16 01 00 000001     TEST3      LA,U      A1,1              . +1 TO A1
129.         000321  74 05 01 00 0 000322               HKJ       1,$+1             . X0 SHOULD BE +1
130.         000322  10 16 01 00 777776                LA,U      A1,0777776        . -1 TO A1
131.         000323  74 05 01 00 0 000324               HKJ       1,$+1             . X0 SHOULD BE 000000777776
132.         000324  10 17 01 00 000001                LA,XU     A1,1              . +1 TO A1
133.         000325  74 05 01 00 0 000326               HKJ       1,$+1             . X0 SHOULD BE +1
134.         000326  10 17 01 00 777776                LA,XU     A1,0777776        . -1 TO A1
135.         000327  74 05 01 00 0 000330               HKJ       1,$+1             . X0 SHOULD BE -1
136.         000330  05 00 00 00 0 000024               SZ        024               . 0 TO A10
137.         000331  50 00 00 00 0 000024               TZ        024               . TEST A0 (SHOULD BE ZERO)
138.         000332  74 05 10 00 0 000333               HKJ       8,$+1             . TZ OR SZ FAILED
139.         000333  52 17 10 00 000000                TE,XU     A8,0              .
140.         000334  74 05 10 00 0 000335               HKJ       8,$+1             . TE (OPERAND IS ZERO) FAILED
141.         000335  52 17 01 00 777776                TE,XU     A1,0777776        .
142.         000336  74 05 10 00 0 000337               HKJ       8,$+1             . TE (-1) FAILED
143.         000337  53 17 01 00 777776                TNE,XU    A1,0777776        .
144.         000340  74 04 00 00 0 000342               J         $+2               . CONTINUE
145.         000341  74 05 10 00 0 000342               HKJ       8,$+1             . TNE FAILED
146.         000342  10 01 10 00 0 000320               LA,H2     A8,TEST3          . +1 TO A8
147.         000343  14 17 10 00 777776                AA,XU     A8,0777776        . SUBTRACT 1
148.         000344  50 00 00 00 0 000024               TZ        A8                .
149.         000345  74 05 10 00 0 000320               HKJ       8,TEST3           . LOAD OR ADD FAILED
150.         000346  74 04 00 00 0 000320               J         TEST3             . START OVER
151.         000347                                     RES       0350-$            .
```

```
                                                . CORE TAG TEST
                                                . ************
                                                . THIS TEST CHECKS THE ADDRESS CAPABILITY OF THE MEMORY BY
                                                . WRITING THE ADDRESS INTO THE MEMORY LOCATION HAVING
                                                . THE SAME ADDRESS.
                                                . TEST IS WRITTEN TO WORK ON 262 K OF MEMORY
                                                . TEST CAN BE LOADED WITH ANY MSR SELECT
                                                . STOP KEY 1 = RECEIVED ADDRESS IN X0
                                                . STOP KEY 2 = EXPECTED ADDRESS + 1 IN X0
                                                . STOP KEY 3 = END OF TEST
                                                . STOP KEY 4 = ERROR STOP, PSR IN X0
01   000350  10 16 02 00 002000      TEST4     LA,U      A2,02000          .
     000351  46 16 16 00 000001                LXI,U     A2,1              . X16 IS 1/02000
     000352  06 00 16 16 2 000000    WR        SX        A2,0,*A2          . STORE TAG
     000353  52 00 02 00 0 000366              TE        A2,LASTAD         . TEST FOR LAST ADDRESS
     000354  74 04 00 00 0 000352              J         $-2               . NO  REPEAT
     000355  26 16 16 00 002000                LXM,U     A2,02000          . YES  X16 IS 1/02000
     000356  10 01 01 16 0 000000    RD        LA,H2     A1,0,A2           . LOAD TAG
     000357  52 16 01 16 2 000000              TE,U      A1,0,*A2          . (A1) EQUALS (A2)
     000360  74 05 10 00 0 000364              HKJ       8,ERDISP          . NOT EQUAL -- GO TO DISPLAY
     000361  52 00 02 00 0 000366              TE        A2,LASTAD         . TEST FOR LAST ADDRESS.
     000362  74 04 00 00 0 000356              J         RD                .
     000363  74 05 04 00 0 000350              HKJ       4,TEST4           . STOP 3 = END OF TEST
     000364  74 05 01 00 0 000365    ERDISP    HKJ       1,$+1             . DISPLAY A1 IN X0
     000365  74 05 02 00 0 000356              HKJ       2,RD              . DISPLAY A2 IN X0
     000366  000001 777776           LASTAD    +         1,0777776         .
     000367                                     RES       0400-$            .
                                                .
                                                . REPEAT INSTRUCTION USED TO CLEAR AND CHECK MEMORY
                                                . *************************************************
                                                . THIS TEST CLEARS THE MEMORY AND CHECKS IF ALL LOCATIONS
                                                . ARE ACTUALLY ZERO.
                                                . TEST WORKS ONLY IN MEMORY 1 (MSR = 0)
                                                . TEST IS WRITTEN TO WORK ON 262 K OF MEMORY
                                                . STOP KEY 1 = END OF TEST
                                                . STOP KEY 4= ERROR. ADDRESS DISPLAYED IN X0
     000400  10 16 01 00 002000      REPINS    LA,U      A1,02000          . START ADDR. TO A1
     000401  05 00 00 00 0 000014              SZ        A0                . ZERO TO A0
     000402  46 16 15 00 000001                LXI,U     A1,1              . A1 IS 1/2000 = DESTINATION
     000403  27 16 01 00 000014                LX,U      X1,014            . X1 IS 0/014 = SOURCE
     000404  23 16 01 00 775776                L,U       R1,0775776        . REPEAT COUNT
     000405  22 00 15 01 2 000000              BT        A1,0,*X1          . CLEAR MEMORY
     000406  26 16 15 00 002000                LXM,U     A1,02000          . RESET START ADDR.
     000407  23 16 01 00 775776                L,U       R1,0775776        . RESET REPEAT COUNT
     000410  63 00 00 15 2 000000              SNE       A0,0,*A1          . SEARCH FOR NON ZEROS
     000411  74 05 01 00 0 000400              HKJ       1,REPINS          . NORMAL END
     000412  74 05 10 00 0 000400              HKJ       8,REPINS          . NON ZERO DETECTED, ADDR. IN X0
     000413                                     RES       0420-$            .
```

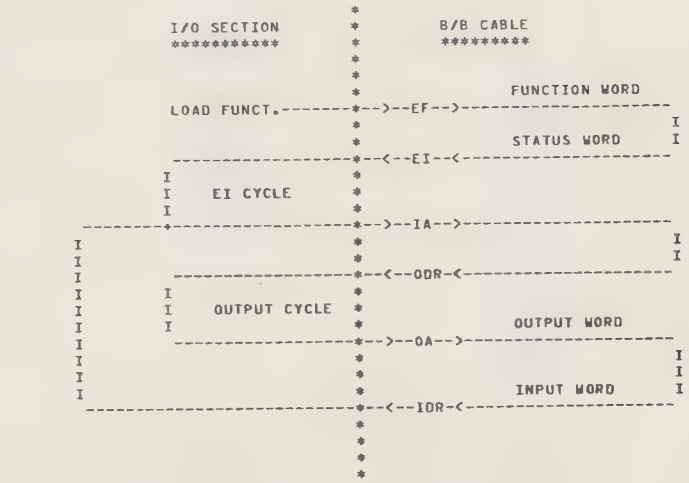
```
                                                . CORE MEMORY TEST
                                                . ***************
                                                . THIS IS APPROX. THE SAME TEST AS IN THE HANDBOOK
                                                . TEST IS WRITTEN TO WORK ON 262 K OF MEMORY
                                                . STOPS:  0              NOT USED
                                                .         1              ERROR, ADDRESS IN XO
                                                .         2              ERROR, PATTERN RECEIVED IN XO
                                                .         3              END OF WRITE
                                                .         4              END OF READ
   01   000420  27 00 01 00 0 000437   TEST5    LX       X1,STARTAD      . X1 IS START ADDR.
        000421  23 00 01 00 0 000440            LR       R1,COUNTAD      . R1 IS NUMBER OF ADDR.
        000422  10 00 00 00 0 000441            LA       A0,PATT         . A0 IS PATTERN
        000423  01 00 00 01 2 000000            SA       A0,0,*X1        . STORE PATTERN TO MEMORY
        000424  70 101 00 0 000423              JGD      R1,$-1          . REPEAT UNTIL R1 IS ZERO
        000425  74 05 04 00 0 000426            HKJ      4,$+1           . END OF WRITE
        000426  27 00 01 00 0 000437            LX       X1,STARTAD      . RELOAD X1
        000427  23 00 01 00 0 000440            LR       R1,COUNTAD      . RELOAD R1
        000430  10 00 02 01 2 000000            LA       A2,0,*X1        . READ MEMORY
        000431  52 00 02 00 0 000441            TE       A2,PATT         . TEST IF PATTERN ARE STILL THE SAME
        000432  74 05 02 00 0 000435            HKJ      2,$+3           . DATA ERROR  DATA IN XO
        000433  70 101 00 0 000430              JGD      R1,$-3          . REPEAT UNTIL R1 IS ZERO
        000434  74 05 10 00 0 000420            HKJ      8,TEST5         . END OF READ
        000435  10 00 01 00 0 000001            LA       A1,01           . READ BAD ADDRESS
        000436  74 05 01 00 0 000433            HKJ      1,$-3           . DISPLAY ADDR. IN XO
        000437  000101 002000          STARTAD  +        1,02000         . FIRST ADDRESS
        000440  000000775776          COUNTAD  +        0775776         . NUMBER OF ADDR. FOR TEST
        000441  777777777777          PATT     +        0777777777777   . PATTERN
        000442                                   RES      0500-$          .
```

```
230.
231.                                                    . ANOTHER MEMORY TEST
232.                                                    . ******************
233.                                                    . TEST IS WRITTEN TO WORK ON 262 K OF MEMORY
234.                                                    . JUMP KEY NONE              WRITE ALL BITS
235.                                                    . JUMP KEY 1                 READ
236.                                                    . JUMP KEY 2                 COMPARE
237.                                                    . JUMP KEY 3                 ALTERNATE WRITE READ AND COMPARE
238.                                                    . JUMP KEY 5                 PATTERN ALL ZEROS
239.                                                    . JUMP KEY 6                 PATTERN 525252525252
240.                                                    . JUMP KEY 7                 PATTERN 252525252525
241.                                                    . STOP KEY 1                 END OF TEST
242.                                                    . STOP KEY 2                 ERROR. DISPLAY ADDRESS IN XO
243.                                                    . STOP KEY 3                 ERROR. DISPLAY DATA IN XO
244.    01   000500  10 JC 01 00 0 000532   TEST6   LA        A1,FUNC          . LOAD THREE FUNCTIONS
245.         000501  74 04 03 00 0 000504            JK        3,$+3            . JUMP ON KEY 3
246.         000502  74 04 02 00 0 000505            JK        2,$+3            . JUMP ON KEY 2
247.         000503  74 04 01 00 0 000506            JK        1,$+3            . JUMP ON KEY 1
248.         000504  73 00 01 00 0 000014            SSC       A1,12            . SHIFT FUNCTION 12 PLACES
249.         000505  73 00 01 00 0 000014            SSC       A1,12            . SHIFT FUNCTION 12 PLACES
250.         000506  73 00 01 00 0 000014   SH12     SSC       A1,12            . SHIFT FUNCTION 12 PLACES
251.         000507  01 07 01 00 0 000522            SA,T1     A1,OPER          . STORE RIGHT MOST FUNCT. TO OPER
252.         000510  23 01 04 00 0 000534            LR,H2     R4,COUNTA        . R1 IS NUMBER OF ADDR.
253.         000511  27 00 16 00 0 000533            LX        A2,STARTA        . X16 IS START ADDR.
254.         000512  74 02 01 00 0 000522            JP        A1,OPER          . JUMP OVER PATTERN SELECT IF NO WRITE
255.         000513  10 00 00 00 0 000535            LA        A0,NUMA          . SELECTED PATTERN TO A0
256.         000514  74 04 07 00 0 000522            JK        7,OPER           . TAKE PATTERN
257.         000515  11 00 00 00 0 000014            LN        A0,014           . INVERT PATTERN
258.         000516  74 04 06 00 0 000522            JK        6,OPER           . TAKE INVERTED PATTERN
259.         000517  05 00 00 00 0 000014            SZ        A0               . PATTERN IS ZERO
260.         000520  74 04 05 00 0 000522            JK        5,OPER           . TAKE ZERO PATTERN
261.         000521  11 00 00 00 0 000014            LN        A0,014           . INVERT ZEROS
262.         000522  74 16 00 16 2 000001   OPER     NOP       0,1,*A2          . THIS WILL BECOME A RADOM FUNCTION
263.         000523  70 104 00 0 000522              JGD       R4,$-1           . REPEAT
264.         000524  54 00 02 00 0 000534            TNG       A2,COUNTA        . TEST FOR NUMBER OF ADDR.
265.         000525  74 05 02 00 0 000530            HKJ       2,$+3            . ERROR ON COMPARE DATA JUMP 2 OR 3
266.                                                                           . DISPLAY FAILURE ADDR. IN XO
267.         000526  74 04 03 00 0 000506            JK        3,SH12           . NEXT FUNCTION ON KEY 3
268.         000527  74 15 01 00 0 000500            HKJ       1,TEST6          . END OF TEST
269.         000530  10 00 04 16 0 000000            L         A4,0,A2          . LOAD A4 FOR DISPLAY
270.         000531  74 05 04 00 0 000522            HKJ       4,OPER           . DISPLAY DATA
271.         000532  5301010000100          FUNC     +         0530010000100    . T1 = TNE,  T2 = LA, T3 = SA.
272.         000533  000001 002000           STARTA   +         1,02000          . START ADDR.
273.         000534  000001 775775           COUNTA   +         1,0775775        . NUMBER OF ADDR. MINUS 2
274.         000535  252525252525           NUMA     +         0252525252525    . PATTERN
275.         000536                           RES      0540-$           .
```

```
                                        . P REGISTER CHECKOUT
                                        . ********************
                                        . THIS ROUTINE TESTS THE P-ADDER AND JUMP LOGIC.
                                        . FIRST ALL LOCATIONS, STARTING AT 2000, WILL BE FILLED WITH
                                        . S L J INSTR. WHICH CAPTURE P INTO LOC. N O P A .
                                        . NEXT INSTR. AFTERWARDS IS N O P A + 1 .
                                        . THE CAPTURED P IS COMPARED TO A CALCULATED ONE.
                                        . STOP KEY 1                ERROR -XO IS EXPECTED P
                                        . STOP KEY 2                ERROR -XO IS CAPTURED P
                                        . STOP KEY 3                END OF TEST
                                        . STOP KEY 4                END OF WRITE
01  000540  27 00 05 00 0 000562  TEST7  LX        X5,SLJINS         . X5 IS SLJ INSTR.
    000541  27 00 15 00 0 000561         LX        A1,FIRSTA         . X15 IS START ADDR.
    000542  06 00 05 15 2 000000         SX        X5,0,*A1          . STORE INSTR. TO CORE
    000543  52 00 01 00 0 000560         TE        A1,LASTA          . TEST FOR LAST ADDR.
    000544  74 04 00 00 0 000542         J         $-2               . REPEAT
    000545  74 05 10 00 0 000556         HKJ       8,PREPA           . ALL WRITTEN
    000546  74 06 00 00 0 000000  NOPA   NOP                         . THIS LOC WILL HOLD P
    000547  10 01 02 00 0 000546         LA,H2     A2,NOPA           . CAPTURED P TO A2
    000550  53 00 01 00 0 000560         TNE       A1,LASTA          . TEST FOR LAST ADDR.
    000551  74 05 04 00 0 000556         HKJ       4,PREPA           . WAS LAST ADDR.
    000552  53 16 02 15 0 000000         TNE,U     A2,0,A1           . TEST A2 = A1
    000553  74 04 00 15 2 000000         J         0,*A1             . YES JUMP
    000554  74 05 02 00 0 000555         HKJ       2,$+1             . NO -XO IS CAPTURED P
    000555  74 05 01 00 0 000540         HKJ       1,TEST7           . NO -XO IS EXPECTED P
    000556  26 16 15 00 002001  PREPA    LXM,U     A1,02001          . PREPARE FOR RESTART
    000557  74 04 00 00 0 002000         J         02000             . JUMP INTO THE CORE AREA
    000560  000001 777776       LASTA   +         1,0777776         . LAST ADDR.
    000561  000001 002000       FIRSTA  +         1,02000           . FIRST ADDR.
    000562  72 01 00 00 0 000546  SLJINS  SLJ       NOPA              . INSTR. TO FILL THE MEMORE
    000563                       RES       0600-$                    .

                                        . TRICK PROGRAM
                                        . *************
                                        . THIS PROGRAM IS TO AVOID MANUAL, OCTAL ARITHMETIC.
                                        . IT CAN BE USED TO TROUBLESHOOT MARGINAL PROBLEMS.
                                        . OPERATION:
                                        . THIS EXAMPLE WILL TEST A MULTIPLY INTEGER. THE RESULT IS UNKNOWN.
                                        . TO SET UP THE LOOP, SET JUMP KEY 1, TURN THE MARGINS, SO THAT
                                        . THE C P U WILL WORK PROPERLY AND START. THEN RELEASE JUMP 1
                                        . AND TURN THE MARGINS UNTIL THE C P U FAILES.
                                        . STOP 1 DISPLAYS WRONG FIRST WORD IN XO
                                        . STOP 2 DISPLAYS WRONG SECOND WORD IN XO
                                        . STOP 3 = RESULT IS EQUAL
    000600  10 00 01 00 0 000611  TRICK   LA        A1,OPE1           . LOAD FIRST OPERAND
    000601  30 00 01 00 0 000612         MI        A1,OPE2           . MULTIPLY
    000602  74 04 01 00 0 000604         JK        1,$+2             . JUMP 1 TO STORE RESULT
    000603  74 04 00 00 0 000605         J         $+2               . NO JUMP 1 TO COMPARE
    000604  71 12 01 00 0 000613         DS        A1,STORAGE        . STORE THE RESULT
    000605  71 17 01 00 0 000613         DTE       A1,STORAGE        . TEST FOR EQUAL
    000606  74 05 01 00 0 000610         HKJ       1,$+2             . UNEQUAL. DISPLAY FIRST WORD
    000607  74 05 04 00 0 000600         HKJ       4,TRICK           . RESULT IS EQUAL
    000610  74 05 02 00 0 000600         HKJ       2,TRICK           . UNEQUAL. DISPLAY SECOND WORD
    000611  012345676543       OPE1    +         012345676543      . OPERAND 1
    000612  765432101234       OPE2    +         0765432101234     . OPRAND 2
    000613                     STORAGE RES       2                 . RESERVE 2 LOCATIONS
    000615                             RES 0620-$                    .
```

```
. I/O BACK TO BACK TESTING IN ISI MODE
. ************************************
. THE ISI BACK TO BACK TESTS USE A JUMPER CABLE TO CONNECT THE INPUT TO THE
. OUTPUT OF THE SAME CHANNEL.
. THE SHORT AS WELL AS THE LONG ISI BACK TO BACK TEST WILL GENERATE THE
.
. FOLLOWING INTERFACE SIGNALS:
.
```




```
.
. DESCRIPTION: THE ISI FUNCTION IS SENT OUT PRODUCING AN EXTERNAL INTERRUPT
. ON THE INPUT SIDE. AN EI IS ANSWERD BY AN IN ACK.
. THIS IA IS ROUTED THROUGH THE CABLE AND BECOMES AN ODR.
. THE ODR CAUSES AN OUTPUT TRANSFER AND THE WORD IS SENT WITH AN OUT ACK.
. THE OA BECOMES AN IOR WHICH IS HANDLED THE SAME WAY AS THE FIRST ONE.
. THE TRANSFER STOPS IF THE INPUT OR OUTPUT ACW HAS A WORD COUNT OF ZERO.
.
```

I/O CABLE WIRING

OUTPUT		INPUT	
EF	CJ/CN	EI	CJ/CN
OA	BU/BY	IDR	BU/BY
IA	BT/BX	ODR	BT/BX
CLEAR	CA/CE	CLEAR	CA/CE

INPUT AND OUTPUT BITS

0	A/E	18	*P/*T
1	B/F	19	*Q/*U
2	C/H	20	*V/*Z
3	D/J	21	*W/AA
4	K/P	22	*X/AB
6	M/S	24	AD/AJ
7	N/T	25	AE/AK
8	UOY	26	AF/AL
9	V/Z	27	AH/AM
10	W/*A	28	AN/AT
11	X/*B	29	AP/AU
12	*C/*H	30	AR/AV
13	*D/*I	31	AS/AW
14	*T/*J	32	AX/BB
15	*G/*K	33	AY/BC
16	*M/*R	34	AZ/BD
17	*N/*S	35	BA/BE

NOTE: * STANDS FOR SMALL LETTERS

```
                                        . SHORT I/O BACK TO BACK TEST
                                        . ****************************
                                        . TEST WORKS ONLY IN MEMORY 1 (MSR = 0)
                                        . STOP KEY 0 = MAKE P EQUAL BACK TO BACK CHANNEL
                                        . STOP KEY 1 = END OF TEST
                                        . THE FUNCTION WORD CAN BE INSPECTED IN LOC. 0200
                                        . THE INPUT AND OUTPUT BUFFER IS AT LOC. 02000
01   000620  74 15 00 00 0 000621  TEST8   HJ      $+1             . ZERO STOP
                                                                    . MAKE P-REG EQUAL B/B CHANNEL NUMBER
     000621  74 13 14 00 0 000622          LMJ     A0,$+1          . THE L M J CAPTURES THE CHANNEL NUMBER
     000622  73 16 00 00 0 000014          LCR     A0              . LOAD CHANNEL SELECT REG.
     000623  10 12 00 00 0 000637          LA      A0,AAIJIN       .
     000624  01 10 00 00 0 000223          SA      A0,0223         . PREPARE INTERRUPT LOCATION
                                                                    . THE ABOVE 5 INSTRUCTIONS ARE ONLY NEEDED IF THE PROGRAM
                                                                    . IS GENERATED ON A BOOTABLE TAPE.
                                                                    .
                                    . ***************************************************************************
                                                                    . MANUAL PROGRAM LOADING STARTS AT LABEL TEST8A.
                                                                    . LOC 0223 MUST CONTAIN INSTR. FROM LABEL AAIJIN
                                                                    . PRESS B/B CHANNEL INTO CHAN. SEL. REG.
     000625  75 14 00 00 0 000000  TEST8A  AACI                     . ALLOW CHANNEL INTERRUPTS
     000626  75 10 00 00 0 000634          LIC     0,BBACW         . ACTIVATE INPUT CHANNEL
     000627  75 10 00 00 0 000635          LFC     0,BBFACW        . SEND FUNCTION
     000630  75 14 00 00 0 000634          LOC     0,BBACW         . ACTIVATE OUTPUT CHANNEL
     000631  75 16 00 00 0 000631  OPBUSY  JOC     0,$             . JUMP ON OUTPUT ACTIVE
     000632  74 06 00 00 0 000000  NOPIN   NOP                     . DUMMY INSTRUCTION
     000633  74 05 01 00 0 000625          HKJ     1,TEST8A        . END OF TRANSFER
     000634  000100 002000         BBACW   +       0100,02000      . OUTPUT AND INPUT A C W
     000635  000001 000636         BBFACW  +       1,BBF           . FUNCTION A C W
     000636  007700770077         BBF     +       007700770077    . FUNCTION WORD
     000637  74 17 00 00 0 000631  AAIJIN  AAIJ    OPBUSY          . THIS INST. HAS TO GO INTO LOC. 0223
                                                                    . INPUT AND OUTPUT BUFFER ARE THE SAME.
                                                                    . PATTERN CAN BE SELECTED BY DISABLE Z TO V7,8.
                                                                    . LOAD PATTERN INTO V7 OR V8. OR SIMPLY BY PRESSING
                                                                    . BITS INTO V7,8 WHILE THE TEST IS RUNNING.
                                    . ***************************************************************************
                                                                    .
                                                                    . IF THE PROGRAM DOES NOT REACH STOP 1 IT HANGS ON
                                                                    . OUTPUT CHANNEL BUSY. TO ALLOW SCOPING THE LOOP HAS TO RUN
                                                                    . CONTINUOUSLY. THE FOLLOWING INSTRUCTIONS SHOULD BE
                                                                    . ADDED INSTEAD OF THE J O C $ AND THE NOOP.
     000640  23 16 04 00 000040          L,U     R4,040          . DELAY COUNT
     000641  70 104 00 0 000632          JGD     R4,NOPIN        . DELAY LOOP
     000642                              RES     0700-$          .
```

```
460.                                        . LONG INPUT OUTPUT BACK TO BACK TEST
461.                                        . ************************************
462.                                        . TEST WORKS ONLY IN MEMORY 1 (MSR = 0)
463.                                    . STOP 0: MAKE P-REG = B/B CHANNEL   . STOP 1: ERROR -SEND WORD IN X0
464.                                    . STOP 2: ERROR - INPUT WORD IN X0   . STOP 4: ERROR - STATUS IN X0
465.                                    . JUMP 1: NO PATTERN SHIFT           . JUMP 2: NO PATTERN CHANGE
466.                                    . JUMP 3: RESELECT CHANNEL
467.
468.  01  000700  10 30 00 00 0 000756  TEST9   LA      A0,JUMPIN            .
469.      000701  01 30 00 00 0 000223          SA      A0,0223              . PREPARE EXT. INTERRUPT
470.      000702  74 15 00 00 0 000703  SELCH   HJ      $+1                  . 0 STOP **MAKE P-REG. = B/B CHANNEL
471.      000703  74 13 14 00 0 000704          LMJ     A0,$+1               . LOADS CHANNEL NUMBER INTO A0
472.      000704  73 16 00 00 0 000014          LCR     A0                   . CHANNEL NUMBER TO CHANNEL SELECT REG.
473.      000705  46 16 01 00 000001           LXI,U   X1,1                 . X1 INCREMENT IS 1
474.      000706  26 16 01 00 000000  PREPAR  LXM,U   X1,0                 . X1 MODIFIER IS ZERO
475.      000707  23 16 04 00 000004          L,U     R4,4                 . PATTERN COUNT IN R4 IS 4
476.      000710  23 16 05 00 000044          L,U     R5,36                . SHIFT COUNT IN R5 IS 36
477.      000711  74 14 03 00 0 000702  GOOVER  JK      3,SELCH              . JUMP KEY 0 TO SELECT CHANNEL
478.      000712  74 14 02 00 0 000717          JK      2,SHPATT             . JUMP KEY 2 KEEP DATA
479.      000713  70 104 00 0 0 000715          JGD     R4,$+2               . JUMP FOR NEXT PATTERN
480.      000714  74 14 00 00 0 000706          J       PREPAR               . OTHERWISE START AGAIN WITH FIRST PATTERN
481.      000715  10 10 04 C1 2 000752          L       A4,PATTER,*X1        . LOAD PATTERN
482.      000716  01 10 04 00 0 000762          S       A4,OPBUF             . STORE PATTERN TO OUTPUT BUFFER
483.      000717  74 14 01 00 0 000725  SHPATT  JK      1,BTOB               . JUMP KEY 1  DON,T SHIFT DATA
484.      000720  70 105 00 0 000722          JGD     R5,$+2               . JUMP FOR NEXT SHIFT
485.      000721  74 14 00 00 0 000710          J       PPEPAR+2             . START WITH UNSHIFTED PATTERN
486.      000722  10 00 04 00 0 000762          L       A4,OPBUF             . OUTPUT BUFFER TO A4
487.      000723  73 00 04 00 0 000001          SSC     A4,1                 . SHIFT IT ONE PLACE
488.      000724  01 00 04 00 0 000762          S       A4,OPBUF             . STORE IT BACK TO OUTPUT BUFFER
489.      000725  75 14 00 00 0 000000  BTOB    AACI                         . ALLOW ALL INTERRUPTS
490.      000726  74 07 00 00 0 000727          AAIJ    $+1                  .
491.      000727  75 10 00 00 0 000760          LIC     0,IPACW              . LOAD INPUT IN CHANNEL
492.      000730  75 10 00 00 0 000761          LFC     0,FUACW              . SEND FUNCTION
493.      000731  75 04 00 00 0 000757          LOC     0,OPACW              . LOAD OUTPUT IN CHANNEL
494.      000732  75 16 00 00 0 000732          JOC     0,$                  . WAIT FOR THE CHANNEL TO CLOSE
495.      000733  10 00 01 00 0 000020          LA      A1,A4                . TRANSFER A4 TO A1 FOR DISPLAY
496.      000734  10 00 02 00 0 000763          LA      A2,IPBUF             . RECEIVED WORD TO A2
497.      000735  53 10 01 00 0 000016          TNE     A1,A2                . ARE THEY EQUAL
498.      000736  74 04 00 00 0 000741          J       $+3                  . YES DO NEXT TEST
499.      000737  74 15 01 00 0 000740          HKJ     1,$+1                . NO - DISPLAY SEND WORD
500.      000740  74 15 02 00 0 000711          HKJ     2,GOOVER             . DISPLAY RECEIVED WORD
501.      000741  53 00 04 00 0 000200          TNE     A4,0200              . TEST STATUS EQUALS FUNCTION
502.      000742  74 04 00 00 0 000711          J       GOOVER               . O.K. START AGAIN
503.      000743  74 05 04 00 0 000744          HKJ     4,$+1                . NO -DISPLAY SEND WORD
504.      000744  10 00 10 00 0 000200          L       A8,0200              . LOAD STATUS FOR DISPLAY
505.      000745  74 15 10 00 0 000711          HKJ     8,GOOVER             . DISPLAY STATUS
506.      000746  74 16 00 00 0 000000  EIIN    NOP                          . INTERRUPT ENTRY
507.      000747  72 15 00 00 0 000000          LPS     0                    . RELOAD PSR
508.      000750  74 16 00 00 0 000000          NOP                          .
509.      000751  74 17 00 00 1 000746          AAIJ    *EIIN                . RETURN TO INTERRUPTED PROGRAM
510.      000752  000700000001         PATTER  +       1                    . PATTERN
511.      000753  777777777776                 +       0777777777776        .
512.      000754  252525252525                 +       0252525252525        .
513.      000755  070707070707                 +       070707070707         .
514.      000756  72 31 00 00 0 000746  JUMPIN  SLJ     EIIN                 . THIS INSTR. WILL GO TO 0223
515.      000757  000001 000762       OPACW   +       1,OPBUF              .
516.      000760  000001 000763       IPACW   +       1,IPBUF              .
517.      000761  000001 000762       FUACW   +       1,OPBUF              .
518.      000762  000000000000       OPBUF   +       0                    .
519.      000763  000000000000       IPBUF   +       0                    .
520.      000764                             RES     01000-$              .
```

```
                        . E S I BACK TO BACK INPUT TEST
                        . ******************************
                        .
                        . THIS TEST REQUIRES AN E S I CHANNEL CONNECTED TO AN
                        . I S I CHANNEL IN THE OTHER CHANNEL GROUP (CROSSWIRED).
                        .
                        . FOR HALF WORD OPERATION START MANUALLY AT LABEL H W O R D .
                        . FOR QUATER WORD OPERATION START MANUALLY AT LABEL Q W O R D .
                        . IF TEST 10 IS ON A BOOT TAPE THE FIRST 7 INSTRUCTIONS CAN BE USED
                        . TO PARAMETERIZE THE TEST. AT THE FIRST 0 STOP ENTER THE
                        . E S I CHANNEL NUMBER IN OCTAL INTO THE P-REG. AT THE SECOND 0 STOP
                        . ENTER THE I S I CHANNEL NUMBER.
                        . JUMP KEY 15 CAN BE USED TO SELECT QUATER WORD.
                        .
                        . THIS TEST WILL GENERATE THE FOLLOWING INTERFACE SIGNALS:
                        .
                        .
                        . ************                                    ************
                        .           *                                    *
                        .           *                                    *
                        .           *        STATUS                      FUNCTION    *
                        .           *-<-EI--<-------------------------------<-EF--<-*
                        .           *                                                *
                        .           *->-IA-->-------------------------------->-ODR->-*
                        .     E S I *                                                * I S I
                        .           *        DATA +EXT. INDEX            O/P WORD    *
                        .           *-<-IDR-<-------------------------------<-OA--<-*
                        .   CHANNEL *                                                * CHANNEL
                        .           *->-IA-->-------------------------------->-ODR->-*
                        .           *                                                *
                        .           *        DATA +EXT. INDEX            O/P WORD    *
                        .           *-<-IDR-<-------------------------------<-OA--<-*
                        .           *                                                *
                        . ************                                    ************
                        .
                        . DESCRIPTION: THE ISI FUNCTION CAUSES AN ESI INTERRUPT. THE FUNCTION WORD
                        . BECOMES THE STATUS AND IS TRANSFERED TO LOCATION 0200. THE ESI CHANNEL
                        . RESPONSE IS AN IN ACKN. WHICH WILL GENERATE AN ODR ON THE ISI CHANNEL.
                        . THE ISI CHANNEL SENDS AN OUTPUT WORD WITH THE CHARACTER IN THE UPPER HALF
                        . AND THE EXTERNAL INDEX IN THE LOWER HALF(SAME AS THE CTMC).
                        . THE ISI OUT ACKN. ARRIVES IN THE ESI CHANNEL AS AN IDR WHICH STARTS
                        . AN ESI INPUT DATA TRANSFER. THE FOLLOWING ESI IN ACKN. WILL START
                        . THE NEXT TRANSFER UNTIL ONE CHANNEL GOES INACTIVE.
                        .
                        . THIS TEST WORKS ONLY IN MEMORY 1 (MSR = 0)
                        .
                        . STOP KEY 1 = ERROR. BAD WORD IN X0
                        . STOP KEY 2 = END OF TEST
                        . JUMP KEY 15 = QUARTER WORD MODE
                        .
01   001000  74 05 00 00 0 001001  TEST10   HJ      %+1             . GET E S I CHANNEL
     001001  74 13 14 00 0 001002           LMJ     A0,%+1          . CHANNEL # TO A0
     001002  01 00 00 00 0 001043           SA      A0,ESICHA       . SAVE E S I CHANNEL NUMBER
     001003  74 05 00 03 0 001004           HJ      %+1             . GET I S I CHANNEL
     001004  74 13 14 03 0 001005           LMJ     A0,%+1          . CHANNEL # TO A0
     001005  01 00 00 03 0 001044           SA      A0,ISICHA       . SAVE I S I CHANNEL
     001006  74 14 17 00 0 001012           JK      15,QWORD        . IF JUMP KEY 15 = QUATER WORD
```

```
                                                 /
                                                 . ***************************************************************************
                                                 .
                                                 . E S I BACK TO BACK INPUT TEST
                                                 .
                                                 . MANUAL LOAD STARTS AT LABEL H W O R D .
                                                 .
01  001007  05 30 00 00 0 000001   HWORD    SZ       X1              . HALF WORD. ZERO TO X1
    001010  10 16 00 00 00 000002            LA,U     A0,2            . WORD COUNT IS 2
    001011  74 34 00 00 0 001014             J        STORWC          . JUMP TO STORE WORD COUNT
    001012  27 16 01 00 000001      QWORD    LX,U     X1,1            . QUATER WORD. 1 TO X1
    001013  10 16 00 00 000004               LA,U     A0,4            . WORD COUNT IS 4
    001014  01 32 00 00 0 001045    STORWC   SA,H1    A0,IOPACW       . STORE WORD COUNT TO ISI OP ACW
    001015  10 30 00 01 0 001061             LA       A0,ESIINS       . ESI INTERRUPT INSTRUCTION
    001016  01 30 00 00 0 000227             SA       A0,0227         . STORE INSTRUCTION TO INT. ADDR.0227
    001017  05 30 00 00 0 001054    SENDDA   SZ       EINBUF          . ZERO TO INPUT BUFFER
    001020  10 30 00 00 0 001052             LA       A0,EIPACWS      . LOAD ESI INPUT ACW FROM STORAGE
    001021  01 30 00 00 0 001053             SA       A0,EIPACW       . STORE ACW TO EXT. SPEC. INDEX LOCATION.
    001022  74 37 00 01 0 001023             AAIJ     $+1             .
    001023  75 14 00 01 0 000000             AACI                     .
    001024  73 16 00 00 0 001043             LCR      ESICHA          .
    001025  75 30 00 00 0 003000             LIC      0,0             . ACTIVATE ESI INPUT CHANNEL
    001026  73 16 00 00 0 001044             LCR      ISICHA          .
    001027  75 13 00 00 0 001055             LFC      0,IFUACW        . SEND ISI FUNCTION
    001030  75 34 00 00 0 001045             LOC      0,IOPACW        . ACTIVATE ISI OUTPUT
    001031  75 36 00 01 0 001031             JOC      0,$             . WAIT FOR CHANNEL TO CLOSE
    001032  74 36 00 00 0 000000    XXX      NOP                      . DUMMY INSTRUCTION
    001033  10 30 01 00 0 001054             LA       A1,EINBUF       . RECEIVED WORD TO A1
    001034  52 30 01 01 0 001057             TE       A1,RESULT,X1    . TEST RESULT
    001035  74 35 01 00 0 001017             HKJ      1,SENDDA        . ERROR. DISPLAY BAD WORD IN X0
    001036  74 35 02 00 0 001017             HKJ      2,SENDDA        . NORMAL COMPLETION
    001037  74 36 00 00 0 000000    ESIINT   NOP                      . ESI INTERRUPT ENTRY
    001040  72 15 00 00 0 000000             LPS      0               . RELOAD PSR
    001041  74 36 00 00 0 000000             NOP                      .
    001042  74 37 00 00 1 001037     -       AAIJ     *ESIINT.        . RETURN JUMP
    001043  000000000000           ESICHA   +        0               . E S I CHANNEL NUMBER
    001044  000000000000           ISICHA   +        0               . I S I CHANNEL NUMBER
    001045  000700 001046           IOPACW   +        0,IOPBUF        . I S I OUTPUT A C W
    001046  000111 001053           IOPBUF   +        0111,EIPACW     . I S I OUTPUT BUFFER
    001047  000222 001053                    +        0222,EIPACW     . I S I OUTPUT BUFFER
    001050  000333 001053                    +        0333,EIPACW     . I S I OUTPUT BUFFER
    001051  000444 001053                    +        0444,EIPACW     . I S I OUTPUT BUFFER
    001052  200100 001054           EIPACWS  +        0200100,EINBUF  . E S I INPUT A C W STORAGE
    001053  000000000000           EIPACW   +        0               . EXTERNAL SPECIFIED LOCATION
    001054  000000000000           EINBUF   +        0               . E S I INPUT BUFFER
    001055  000001 001056           IFUACW   +        1,IFUNCT        . I S I FUNCTION A C W
    001056  007007007007           IFUNCT   +        007007007007    . FUNCTION WORD
    001057  000222 000111           RESULT   +        0222,0111       . HALF WORD RESULT
    001060  111222333444                    +        0111222333444   . QUATER WORD RESULT
                                                 . IF THE PROGRAM IS LOADED BY HAND
                                                 . THE FOLLOWING INSTRUCTION SHOULD GO TO 0227
    001061  72 01 00 00 0 001037    ESIINS   SLJ      ESIINT          . CAPTURE P ON INT.
                                                 .
                                                 . ***************************************************************************
                                                 .
                                                 . IF THE LOOP DOES NOT REACH ANY STOP IT HANGS ON THE
                                                 . J O C . TO ENABLE SCOPING REPLACE THE J O C AND N O P WITH
    001062  23 16 04 00 00002D               L,U      R4,020          . DELAY COUNT
    001063  70 104 00 0 001032               JGD      R4,XXX          . DELAY INSTRUCTION
                                                 .
                                                 . STOP 1 ERROR. WRONG WORD IN X0      . STOP 2 NORMAL COMPLETION
                                                 . JUMP 15 SELECT QUATER WORD MODE
    001064                                       RES      01100-$         .
```

15

```
642.
643.                    . E S I OUTPUT BACK TO BACK TEST
644.                    . ******************************
645.
646.  . THIS TEST WILL GENERATE THE FOLLOWING INTERFACE SIGNALS
647.  .
648.  . ************                                                ************
649.  .            *                                                *
650.  .            *          STATUS                  FUNCTION      *
651.  .            *-<-EI--<-------------------------------------<-EF--<-*
652.  .            *                                                *
653.  .            *->-IA-->-------------------                      *
654.  .   E S I    *                                                *   I S I
655.  .            *          FUNCTION                 STATUS       *
656.  .            *->-EF--<-------------------------------------->-EI-->-*
657.  .            *                                                *
658.  .            *-<-ODR-<-------------------------------------<-IA--<-*
659.  .  CHANNEL    *                                                *   CHANNEL
660.  .            *          O/P CHAR.               I/P WORD      *
661.  .            *->-OA--<-------------------------------------->-IDR->-*
662.  .            *                                                *
663.  .            *-<-ODR-<-------------------------------------<-IA--<-*
664.  .            *                                                *
665.  . ************                                                ************
666.  .
667.  . DISCRIPTION: THE ISI FUNCTION TRANSFER IS USED TO LOAD THE OUTPUT BUFFER
668.  . REGISTER(OBR) WITH THE EXTERNAL INDEX. THIS BIAS ADDRESS STAYS IN THE OBR AND
669.  . WILL BE USED THROUGHOUT THE FOLLOWING OUTPUT TEST. THE OUTPUT STARTS WITH AN
670.  . FSI FUNCTION CAUSING AN INTERRUPT ON THE ISI CHANNEL. THE FUNCTION WORD CAN
671.  . BE FOUND IN THE STATUS LOCATION. THE ISI CHANNEL RESPONSE IS AN IN ACKN.
672.  . WHICH GENERATES AN ESI OUTPUT DATA REQUEST (ODR). THE EXTERNAL INDEX USED FOR
673.  . THIS TRANSFER IS THE ONE LOADED WITH THE PREVIOUS ISI FUNCTION TRANSFER.
674.  . THE FOLLOWING ESI OUT ACKN. STARTS THE NEXT TRANSFER.
675.                    . E S I OUTPUT BACK TO BACK TEST
676.                    . ******************************
677.                    . THIS TEST USES DATA OUT OF TEST 10
678.                    . TEST WORKS ONLY IN MEMORY 1 (MSR = 0)
679.                    . STOP 1 = DATA ERROR -DISPLAY BAD WORD
680.                    . STOP 2 = DATA O.K. -DISPLAY GOOD WORD
681.                    . STOP 3 = END OF TEST
682.                    . JUMP 15 = QUARTER WORD MODE
```

```
683.                                        /
684.    01  001100  10 00 00 00 0 001061  TEST1OA   LA      A0,ESIINS       .
685.        001101  01 00 00 00 0 000223            SA      A0,0223         . PREPARE INTERRUPT ADDR.
686.        001102  73 16 00 00 0 001044  SENDIF    LCR     ISICHA          . ISI CHANNEL TO CHANNEL SEL. REG.
687.        001103  75 14 00 00 0 000000            AACI                    .
688.        001104  74 07 00 00 0 001105            AAIJ    $+1             .
689.        001105  75 10 00 00 0 001142            LFC     0,IFUAC         . SEND ISI FUNCTION.
690.                                                                        . THE O B R WILL BE LOADED WITH THE
691.                                                                        . EXTERNAL INDEX.
692.        001106  75 12 00 00 0 001106            JFC     0,$             . WAIT
693.        001107  27 16 01 00 0 000001            L,U     X1,1            . QUARTER WORD X1 IS 1
694.        001110  74 24 17 00 0 001112            JK      15,$+2          . JUMP 15 SAYS QUATER WORD
695.        001111  05 20 00 00 0 000001            SZ      X1              . HALF WORD X1 IS ZERO
696.        001112  10 30 01 00 0 007200            LA      A1,0200         . ISI FUNCTION MUST BE IN STATUS LOC.
697.        001113  52 30 01 00 0 001143            TE      A1,IFUNC        . IS IT EQUAL
698.        001114  74 35 01 00 0 001102            HKJ     1,SENDIF        . NO. DISPLAY BAD STATUS
699.        001115  10 30 00 01 0 001146            LA      A0,EOPACWS,X1   . GET ESI O/P ACW FROM STORAGE
700.        001116  01 30 00 03 0 001153            SA      A0,EOPACW       . STORE IT TO EXT. SPECIFIED LOC.
701.        001117  75 00 00 01 0 001151            LIC     0,IIPACW,X1     . ACTIVATE ISI INPUT
702.        001120  73 16 00 03 0 001043            LCR     ESICHA          . ESI CHANNEL TO CHANNEL SEL. REG.
703.        001121  75 10 00 03 0 001144            LFC     0,EFUACW        . SEND ESI FUNCTION
704.        001122  75 34 00 00 0 000000            LOC     0,0             . ACTIVATE ESI OUTPUT
705.        001123  75 36 00 00 0 001123            JOC     0,$             . WAIT FOR ESI OUTPUT TO CLOSE
706.        001124  74 06 00 00 0 000000  XXXX      NOP                     . DUMMY INSTR.
707.        001125  10 30 01 00 0 000200            LA      A1,0200         . GET STATUS
708.        001126  52 30 01 00 0 001145            TE      A1,EFUNCT       . TEST ESI FUNCTION EQUALS STATUS
709.        001127  74 35 01 00 0 001102            HKJ     1,SENDIF        . BAD STATUS DISPLAY IN X0
710.        001130  27 00 02 00 0 001160            L       X2,ONENUL       .
711.        001131  46 16 01 03 000002              LXI,U   X1,2            .
712.        001132  23 02 04 01 0 001151            L,H1    R4,IIPACW,X1    . GET NUMBER OF INPUT WORDS
713.        001133  70 104 00 3 001135  COMPAR     JGD     R4,$+2          . JUMP UNTIL LAST WORD
714.        001134  74 35 04 00 0 001102            HKJ     4,SENDIF        . END OF TEST
715.        001135  10 30 01 02 2 001154            LA      A1,IIPBUF,*X2   . GET INPUT WORD
716.        001136  10 30 02 00 0 000015            LA      A2,A1           . LOAD A2 TO DISPLAY GOOD WORD
717.        001137  52 30 01 01 2 001161            TE      A1,RESUL,*X1    . EQUAL TO RESULT BUFFER?
718.        001140  74 35 01 00 0 001133            HKJ     1,COMPAR        . NO. DISPLAY BAD WORD IN X0
719.        001141  74 35 02 00 0 001133            HKJ     2,COMPAR        . YES. TEST NEXT WORD
720.        001142  000001 001143          IFUAC     +       1,IFUNC         . ISI FUNCTION ACW
721.        001143  070070 001153          IFUNC     +       070070,EOPACW   . ISI FUNCTION
722.        001144  000001 001145          EFUACW    +       1,EFUNCT        . ESI FUNCTION
723.        001145  070707070707          EFUNCT    +       070707070707    . ESI FUNCTION
724.        001146  000002 001150          EOPACWS   +       2,EOPBUF        . ESI OUTPUT ACW
725.        001147  000004 001150                    +       4,EOPBUF        . ESI OUTPUT ACW
726.        001150  111222333444          EOPBUF    +       0111222333444   . ESI OUTPUT BUFFER
727.        001151  000002 001154          IIPACW    +       2,IIPBUF        . ISI INPUT ACW
728.        001152  000004 001154                    +       4,IIPBUF        . ISI INPUT ACW
729.        001153  000020000030          EOPACW    +       0               . EXTERNAL SPECIFIED INDEX LOCATION
730.        001154  000020000030          IIPBUF    +       0               . ISI INPUT BUFFER
731.        001155  000020000000                    +       0               . ISI INPUT BUFFER
732.        001156  000020000000                    +       0               . ISI INPUT BUFFER
733.        001157  000020000000                    +       0               . ISI INPUT BUFFER
734.        001160  000001 000000          ONENUL    +       1,0             .
735.        001161  111222333444          RESUL     +       0111222333444   . FIRST RESULT IN H-WORD
736.        001162  070707110111                    +       070707110111    . FIRST RESULT IN Q-WORD
737.        001163  111222111222                    +       0111222111222   . SECOND RESULT IN H-WORD
738.        001164  070707110222                    +       070707110222    . SECOND RESULT IN Q-WORD
739.        001165  000020000000                    +       0               .
740.        001166  070707110333                    +       070707110333    . THIRD RESULT IN Q-WORD
741.        001167  000020000000                    +       0               .
742.        001170  111222332444                    +       0111222332444   . FORTH RESULT IN Q-WORD
```

```
743.
744.                                        . CONSOLE 4009 TEST
745.                                        . ******************
746.                                        . THIS TEST WILL TAKE THE CONSOLE INPUT AND DISPLAY IT ON THE SCREEN.
747.    01  001171                         RES       01200-$           .
748.        001200  75 10 17 00 0 001207    TEST12    LFC       15,COEFAC       . SEND C R T SELECT TO CONSOLE
749.        001201  75 12 17 00 0 001201              JFC       15,$            . JUMP ON FUNCTION IN CHANNEL
750.        001202  75 00 17 00 0 001211              LIC       15,COIOAC       . ACTIVATE INPUT CHANNEL
751.        001203  75 02 17 00 0 001203              JIC       15,$            . WAIT FOR INPUT
752.        001204  75 04 17 00 0 001211              LOC       15,COIOAC       . ACTIVATE OUTPUT
753.        001205  75 06 17 00 0 001205              JOC       15,$            . WAIT FOR OUTPUT TO CLOSE
754.        001206  74 05 01 00 0 001202              HKJ       1,TEST12+2      . END OF LOOP
755.        001207  000001 001210           COEFAC    +         1,COFUNC        . FUNCTION A C W
756.        001210  400000 000000           COFUNC    +         0400000,0       . C R T FUNCTION
757.        001211  000001 001212           COIOAC    +         1,COIOBU        . INPUT OUTPUT BUFFER A C W
758.        001212  000000000000           COIOBU    +         0               . I/O BUFFER
759.                                        . CONSOLE 4004 TEST
760.                                        . ******************
761.        001213  75 00 17 00 0 001220    TEST12A   LIC       15,CIOACW       . ACTIVATE INPUT
762.        001214  75 02 17 00 0 001214              JIC       15,$            . WAIT FOR INPUT
763.        001215  75 04 17 00 0 001220              LOC       15,CIOACW       . SEND CHAR. TO CONSOLE
764.        001216  75 06 17 00 0 001216              JOC       15,$            . WAIT FOR OUTPUT TO CLOSE
765.        001217  74 05 01 00 0 001213              HKJ       1,TEST12A       . END OF LOOP
766.        001220  000001 001221           CIOACW    +         1,CIOBUF        . I/O A C W
767.        001221  000000000000           CIOBUF    +         0               . I/O BUFFER
768.                                         . HOW TO WRITE A PROGRAM IN OCTAL INSTRUCTION CODE?
769.                                         . ************************************************
770.                                        . THIS IS AN EXAMPEL
771.                                        F         FORM      6,4,4,4,2,16    . FORM STATEMENT FOR INSTRUCTIONS.
772.                                                  . IT CAN BE CALLED BY F IN COLOMN 11 TO TYPE IN INSTR.
773.                                                  . IN OCTAL FORMAT.
774.                                                  . ALL OCTAL FIELDS MUST START WITH ZERO.
775.        001222                          RES 01230-$                 . FORCES THE START ADDR. TO LOC. 1230
776.        001230  75 10 17 00 0 001237              F         075,010,017,0,0,01237
777.        001231  75 12 17 00 0 001231              F         075,012,017,0,0,01231
778.        001232  75 00 17 00 0 001241              F         075,0,017,0,0,01241
779.        001233  75 02 17 00 0 001233              F         075,02,017,0,0,01233
780.        001234  75 04 17 00 0 001241              F         075,04,017,0,0,01241
781.        001235  75 06 17 00 0 001235              F         075,6,017,0,0,01235
782.        001236  74 15 01 00 0 001232              F         074,5,1,0,0,01232
783.                                                  . AND DATA THIS WAY
784.        001237  000001 001240                     +         1,01240
785.        001240  400000000000                     +         0400000000000
786.        001241  000001 001242                     +         01,01242
787.                                        . THE ABOVE PROGRAM IS THE SAME AS TEST 12
788.        001242                          RES       01300-$
```

```
                                        * C T M C BACK TO BACK TEST
                                        * ************************
                                        * TEST WORKS ONLY IN MEMORY 1 (MSR = 0)
                                        * STOP 1 = END OF TEST
                                        * JUMP KEY 1 = ENABLE DATA COMPARE
                                        * JUMP KEY 2 = DISABLE LOOK FOR SYNCH FUNCTION
                                        * JUMP KEY 3 = RESELECT C T M C CHANNEL
                                        * JUMP KEY 4 = RESELECT C L T
                                        * FIRST ZERO STOP = MAKE P EQUAL CTMC CHANNEL
                                        * SECOND ZERO STOP = MAKE P EQUAL BIAS ADDR. (CTMC AND CLT)
                                        * THIRD ZERO STOP = MAKE P EQUAL 3 OF 7 CODE (FORM 0--0XXX0)
01   001300  74 05 00 00 0 001301   TEST11   HJ        $+1                 *
     001301  74 13 14 00 0 001302            LMJ       A0,$+1              *
     001302  01 30 00 00 0 001341            SA        A0,CCHA             * SAVE C T M C CHANNEL
     001303  74 15 00 00 0 001304            HJ        $+1                 *
     001304  74 13 14 00 0 001305            LMJ       A0,$+1              *
     001305  01 30 00 00 0 001342            SA        A0,CLTID            * SAVE C L T BIAS ADDR.
     001306  74 15 00 00 0 001307            HJ        $+1                 *
     001307  74 13 14 00 0 001310            LMJ       A0,$+1              *
     001310  01 30 00 00 0 001343            SA        A0,CFUNCT           * SAVE C L T FUNCTION CODE. FORM 0XXX0
     001311  74 14 03 03 0 001300   TEST11A  JK        3,TEST11            * JUMP 3--RESELECT CTMC CHANNEL
     001312  74 14 04 04 0 001303            JK        4,TEST11+3          * JUMP 4--RESELECT C L T
     001313  73 16 00 00 0 001341            LCR       CCHA                * LOAD CHANNEL SELECT REGISTER
     001314  27 00 01 00 0 001342            L         X1,CLTID            * X1 IS CTMC AND CLT BIAS
     001315  27 10 02 00 0 001344            L         X2,COPACW           * X2 IS OUTPUT ACW
     001316  27 00 03 00 0 001345            L         X3,CIPACW           * X3 IS INPUT ACW
     001317  06 10 02 01 0 000000            S         X2,0,X1             * STORE O/P ACW TO BIAS ADDR.
     001320  06 00 03 01 0 000001            S         X3,1,X1             * STORE I/P ACW TO BIAS + 1
     001321  75 70 00 00 0 000000            LIC       0,0                 * ACTIVATE INPUT
     001322  23 16 02 00 000007              L,U       R2,07               * FUNCTION MASK
     001323  10 00 00 00 0 001343            LA        A0,CFUNCT           * A0 IS FUNTION  (THREE OF SIX CODE)
     001324  43 16 00 00 000001              MLU,U     A0,1                * PUT SEND BIT INTO FUNCTION
     001325  01 10 01 00 0 001343            SA        A1,CFUNCT           * STORE SEND FUNCTION INTO BUFFER
     001326  75 10 00 00 0 001346            LFC       0,CFUACW            * SEND FUNCTION
     001327  75 12 00 00 0 001327            JFC       0,$                 *
     001330  75 04 00 00 0 000000            LOC       0,0                 * ACTIVATE OUTPUT
     001331  75 16 00 00 0 001331            JOC       0,$                 *
     001332  74 04 01 00 0 001357            JK        1,COMPDA            * IF JUMP KEY 1 GO TO COMPARE DATA
     001333  74 14 02 00 0 001311            JK        2,TEST11A           * IF JUMP KEY 2 DONT SEND LOOK FOR SYNCH
     001334  43 16 40 00 000004     SENLFS   MLU,U     A0,4                * GENERATE LOOK FOR SYNCH FUNCTION
     001335  01 00 01 00 0 001343            SA        A1,CFUNCT           * STORE L F S TO FUNCTION BUFFER
     001336  75 10 00 00 0 001346            LFC       0,CFUACW            * SEND LFS FUNCTION
     001337  75 12 00 00 0 001337            JFC       0,$                 *
     001340  74 05 01 00 0 001311            HKJ       1,TEST11A           * END OF TEST
     001341  000010000010           CCHA     =         010                 * C T M C CHANNEL
     001342  000100002026           CLTID    +         02026               * CTMC + CLT BIAS ADDR.
     001343  000300001210           CFUNCT   +         01210               * FUNCTION, THREE OF SIX CODE. FORM 0***0
     001344  000120 001347           COPACW   +         020,COPBUF          * OUTPUT ACW
     001345  000050 003000           CIPACW   +         050,03000           * INPUT BUFFER IS AT ADDR. 03000
                                                                           * WORD COUNT HAS TO BE BIGGER THAN EXPECT.
     001346  000001 001343           CFUACW   +         1,CFUNCT            * FUNCTION ACW
     001347  000026 000026           COPBUF   +         026,026             * OUTPUT BUFFER
     001350  000001 003026                    +         01,026
     001351  000003 000002                    +         03,02
     001352  000005 000004                    +         05,04
     001353  000007 000006                    +         07,06
     001354  000011 000010                    +         011,010
     001355  000013 000012                    +         013,012
     001356  000015 000014                    +         015,014
                                                                           * INPUT BUFFER IS AT LOCATION 03000
     001357  74 06 00 00 0 000000   COMPDA   NOP                           * HERE YOU CAN START A PROGR. TO COMP DATA
     001360                                  RES       01400-$             *
```

```
                                                /
                                                .  M S A TURNAROUND TEST
                                                .  **********************
                                                .  TEST WORKS ONLY IN MEMORY 1 (MSR = 0)
                                    . JUMP 1 = RESELECT MSA CHANNEL     . JUMP 2 = RESELECT PATTERN
                                    . STOP 1 = WRONG WORD IN XO         . STOP 2 = SECOND WRONG WORD IN XO
                                    . STOP 3 = END OF TEST              . STOP 4 = DISPLAY STATUS IN XO
                                    . STOP 0 (FIRST) MAKE P = MSA CHAN. . STOP 0 (SECOND) MAKE P = PATTERN
01  001400  1D 7C 00 00 C 001444   TEST13   LA        A0,EIINS          .
    001401  01 30 00 00 0 000223             SA        A0,0223           PREPARE INTERRUPT LOCATION
    001402  74 05 00 00 0 001403   CHASEL   HJ        $+1               . PRESS MSA CHANNEL INTO P
    001403  74 13 14 00 0 001404             LMJ       A0,$+1            . SAVE P INTO A0
    001404  01 01 00 00 0 001453             SA,H2     A0,MCHA           . SAVE CHANNEL
    001405  74 05 00 00 0 001406   PATSEL   HJ        $+1               . PRESS PATTERN INTO P
    001406  74 13 14 00 0 001407             LMJ       A0,$+1            . SAVE P INTO A0
    001407  01 30 00 00 0 001455             SA        A0,PTTRN          . SAVE PATTERN
    001410  01 32 00 00 0 001455             SA,H1     A0,PTTRN          . STORE PATTERN INTO SECOND HALF WORD
    001411  73 16 00 00 0 001453             LCR       MCHA              .
    001412  75 14 00 00 0 000000             AACT                        .
    001413  74 04 01 00 0 001402   TURN      JK        1,CHASEL          . JUMP 1-RESELECT CHANNEL
    001414  74 04 02 00 0 001405             JK        2,PATSEL          . JUMP 2-RESELECT PATTERN
    001415  05 00 00 00 0 000200             SZ        0200              . CLEAR STATUS LOC
    001416  75 10 00 00 0 001456             LIC       0,MIPACW          . ACTIVATE INPUT CHANNEL
    001417  75 10 00 00 0 001457             LFC       0,MFUACW          . SEND FUNCTION
    001420  75 12 00 00 0 001420             JFC       0,$               .
    001421  75 04 00 00 0 001460             LOC       0,MOPACW          . ACTIVATE OUTPUT CHANNEL
    001422  51 30 00 00 0 000200             TNZ       0200              . STATUS STILL ZERO?
    001423  74 07 00 00 0 001422             AAIJ      $-1               . WAIT
    001424  10 30 10 00 0 000200             L         A8,0200           . LOAD STATUS
    001425  42 16 10 00 337400               AND,U     A8,0337400        . MASK STATUS
    001426  52 16 11 00 006000               TE,U      A9,06000          . CHAN + DEV END?
    001427  74 05 10 00 0 001436             HKJ       8,MAUX            . NO. GET AUX. STATUS
    001430  71 13 01 00 0 001464             DL        A1,MIPBUF         . LOAD INPUT WORDS
    001431  52 00 01 00 0 001455             TE        A1,PTTRN          . FIRST WORD O.K.
    001432  74 05 01 00 0 001433             HKJ       1,$+1             . NO- DISPLAY IN XO
    001433  53 00 02 00 0 001455             TNE       A2,PTTRN          . SECOND WORD O.K.
    001434  74 05 04 00 C 001413             HKJ       4,TURN            . YES-REPEAT
    001435  74 05 02 00 0 001436             HKJ       2,MAUX            . NO- DISPLAY IN XO
    001436  05 00 00 00 C 000200   MAUX      SZ        0200              . CLEAR STATUS
    001437  75 10 00 00 0 001461             LFC       9,MFAACW          . SEND FUNCTION TO GET AUX. STATUS
    001440  51 30 00 00 0 000200             TNZ       0200              .
    001441  74 07 00 00 0 001440             AAIJ      $-1               . WAIT FOR INTERRUPT
    001442  10 00 10 00 C 000200             L         A8,0200           . AUX. STATUS TO A8
    001443  74 05 10 00 0 001413             HKJ       8,TURN            . DISPLAY AUX STATUS IN XO
    001444  72 01 00 00 0 001445   EIINS     SLJ       MEIHA             . INSTR. FOR INTERRUPT LOC.
    001445  74 07 00 00 0 001445   MEIHA     AAIJ      *                 . RETURN JUMP
    001446  72 14 00 00 0 001454             SCN       0,MCHAS           . GET CHANNEL NUMBER
    001447  10 00 17 00 0 001453             L         A15,MCHA          . GET ORIGINAL CHANNEL
    001450  52 00 17 00 0 001454             TE        A15,MCHAS         . WAS INT. FROM MSA CHANNEL
    001451  05 00 00 00 0 000200             SZ        0200              . NO- CLEAR STATUS
    001452  74 14 00 07 C 001445             J         MEIHA             . RETURN
    001453  000170000003           MCHA      +         3                 . MSA CHANNEL NUMBER
    001454  000100000000           MCHAS     +         0                 . CHANNEL NUMBER STORAGE
    001455  252525252525           PTTRN     +         0252525252525     . PATTERN STORAGE
    001456  000102 001464          MIPACW    +         2,MIPBUF          . INPUT ACW
    001457  000101 001462          MFUACW    +         1,MFUBUF          . FUNCTION ACW
    001460  200136 001455          MOPACW    +         0200006,PTTRN     . OUTPUT ACW
    001461  000101 001463          MFAACW    +         1,MFABUF          . AUX FUNCT. ACW
    001462  000100401367           MFUBUF    +         0401367           . TURNAROUND FUNCTION
    001463  000100000361           MFABUF    +         0361              . AUX. FUNCTION
    001464  000100000000           MIPBUF    +         0                 . INPUT BUFFER
    001465  000100000000                     +         0                 .
    001466                                   RES       07000-$           .                                   ***
```

```
915.                                    /
916.                                    $(3)                    . DEFINE A SECOND I-BANK                ***
917.                                    B                       . B IS THE LABEL FOR THIS BANK          ***
918.                                                            . IT IDENTIFIES THE PROGR. WHICH
919.                                                            . WILL GO INTO THE HIDDEN MEMORY.
920.                                    . N O T E : ALL LABELS USED IN AN INSTRUCTION IN THIS PROGRAM PART
921.                                    . *******   MUST BE DECREMENTED BY B. THIS IS THE AMOUNT THE PROGR. IS
922.                                    . OFFSET FROM ZERO.
923.                                            . ROUTINE TO BUILT P S R AFTER BOOT
924.                                            . **********************************
925.                                                    . P S R IS STORED TO 0207
926.                                                    . BOOT CHANNEL NUMBER IS STORED TO 0206
927.                                                    . M S R  READY FOR INDEXING IS STORED TO I C R 0177
928.      03  000000  72 14 00 00 0 000157         SCN     0,0157          . BOOT CHANNEL TO XA3
929.          000001  10 .0 60 17 0 003040         LA      A0,040,A3       . ACW TO XA0
930.          000002  42 16 00 00 077777          AND,U   A0,077777       . EXTRACT ADDRESS
931.          000003  52 16 01 00 002000          TE,U    A1,02000        . IS IT 02000?
932.          000004  74 15 00 00 0 000004         HJ      $-B             . NO. DISPLAY ACW IN X0
933.          000005  42 16 00 00 7000000         AND,U   A0,0700000      . EXTRACT MSR
934.          000006  01 10 00 03 15 0 000206     SA      A3,0206,A1      . STORE BOOT CHANNEL
935.          000007  01 10 01 00 0 000177         SA      A1,0177         . SAVE MSR INTO 0177
936.          000010  01 11 01 15 0 000207         SA,HZ   A1,0207,A1      . STORE MSR
937.          000011  10 12 02 15 0 000207         LA,S4   A2,0207,A1      . LOAD A2 WITH MSR. FORM 0--0X0
938.          000012  46 12 16 15 0 000207         LXI,S4  A2,0207,A1      . LOAD BI. A2 IS 0--0X0/0--0X0
939.          000013  73 12 02 00 0 000003         LSSL    A2,3            . SHIFT TO FORM FINAL PSR.
940.          000014  01 12 02 15 0 000207         SA      A2,0207,A1      . STORE PSR TO 0207
941.                                            . RESTART POINT   NOTE: DO NOT CLEAR I C R
942.                                            .                 ************************
943.                                            . ***************************************
944.          000015  27 10 13 00 0 000177 RESTART LX      X11,0177        . LOAD M S R FROM 0177
945.          000016  72 15 00 13 0 000207         LPS     BPSR,X11        . LOAD PSR
946.          000017  74 16 00 00 0 000000         NOP                     . NO OPERATION
947.          000020  75 14 00 00 0 000000         AACI
948.                                            . ROUTINE TO SELECT ONE OUT OF 15 TESTS.
949.                                            . ***************************************
950.          000021  74 14 01 00 0 000250 WAITLOOP JK      1,TEST1         . SELECT TEST  1
951.          000022  74 14 02 00 0 000300         JK      2,TEST2         . SELECT TEST  2
952.          000023  74 14 03 00 0 000320         JK      3,TEST3         . SELECT TEST  3
953.          000024  74 14 04 00 0 000350         JK      4,TEST4         . SELECT TEST  4
954.          000025  74 14 05 00 0 000420         JK      5,TEST5         . SELECT TEST  5
955.          000026  74 14 06 00 0 000500         JK      6,TEST6         . SELECT TEST  6
956.          000027  74 14 07 00 0 000540         JK      7,TEST7         . SELECT TEST  7
957.          000030  74 14 10 00 0 000620         JK      8,TEST8         . SELECT TEST  8
958.          000031  74 14 11 00 0 000700         JK      9,TEST9         . SELECT TEST  9
959.          000032  74 14 12 00 0 001000         JK      10,TEST10       . SELECT TEST 10
960.          000033  74 14 13 00 0 001300         JK      11,TEST11       . SELECT TEST 11
961.          000034  74 14 14 00 0 001200         JK      12,TEST12       . SELECT TEST 12
962.          000035  74 14 15 00 0 001400         JK      13,TEST13       . SELECT TEST 13
963. U        000036  74 14 16 00 0 000000         JK      14,TEST14       . SELECT TEST 14
964. U        000037  74 14 17 00 0 000000         JK      15,TEST15       . SELECT TEST 15
965.          000040  74 17 00 00 0 000021         AAIJ    WAITLOOP-B      .
966.                                            . ROUTINE TO IGNORE INTERRUPTS
967.                                            . ****************************
968.                                            . FOR THIS THE LOCATIONS 0210-0250 MUST BE PADDED AS SHOWN
969.          000041  75 14 00 00 0 000000 IGNORE  AACI                    . ALLOW ALL CHANNEL INTERRUPTS
970.          000042  72 15 00 00 0 000000         LPS     0               . RELOAD PSR
971.          000043  74 17 00 01 0 000000         AAIJ    0,X1            . ALLOW ALL INTERRUPTS AND RETURN
972.          000044                               RES     02000-$         . RESERVE DUMP AREA             ***
```

```
    973.                                                /
    974.                                                $(0)                                    . D-BANK  USED TO PRODUCE THE TAPE      ***
    975.                                                BRUTAL          . START ADDR. TO PRODUCE BOOT TAPE                              ***
    976.        00    000000  27 16 01 00 000200        BRUTAL  L,U     X1,A                    . SOURCE IS FIRST I-BANK                ***
    977.              000001  27 16 02 00 000200                L,U     X2,B+0200               . DESTINATION IS IN RES. AREA           ***
    978.              000002  23 16 01 00 001600                L,U     R1,01600                . WORD COUNT IS 01600 WORDS             ***
    979.              000003  46 16 01 00 000001                LXI,U   X1,1                    . INCREMENT IS 1                        ***
    980.              000004  46 16 02 00 000001                LXI,U   X2,1                    . INCREMENT IS 1                        ***
    981.              000005  22 00 02 01 2 000000              BT      X2,0,*X1                . BLOCK TRANSFER                        ***
    982.              000006  10 16 00 00 000011              LA,U    A0,TPKT                 . TAPE PACKAGE ADDR. TO A0              ***
    983. U            000007  72 11 00 00 0 000000            ER      IOW$                    . EXEC REQUEST FOR I/O                  ***
    984. U            000010  72 11 00 00 0 000000            ER      EXIT$                   . END OF PROGRAM TO DUMP A TAPE         ***
    985. U            000011  310625120505            TPKT    I$OT    'TAPE',W$  02000,B      .                                       ***
                      000012  050505050535
                      000013  000000 000000
                      000014  0000 0000 0000
                      000015  0 002000 000000
    986.                      000000000000                    END     BRUTAL                  . START ADDR. FOR EXEC                  ***

UNDEFINED SYMBOLS:
EXIT$       IOW$        TEST15      TEST14      W$

END ASM. ERRORS : NONE
```

```
@MAP,I    M,ABS                                                    .      ***
MAP28R1 RL71-3 09/12/76 16:14:08 (,0)
     1.       IBANK      I1,0                                                .      ***
     2.        IN        B                                                   .      ***
     3.       DBANK      D1                                                  .      ***
     4.       IN      B                                                      .      ***
TEST14        IS NOT DEFINED - REFERENCED IN ELEMENT                 B
TEST15        IS NOT DEFINED - REFERENCED IN ELEMENT                 B


ADDRESS LIMITS
I1                000000 003777       2048 IBANK WORDS DECIMAL
D1                040000 040015         14 DBANK WORDS DECIMAL
STARTING ADDRESS  040000


***************IBANK I1              000000 003777


              SEGMENT $MAIN$          000000 003777

ERU$/SYS72-8
                          $(1)    000000 001777
                          $(3)    002000 003777


***************DBANK D1              040000 040015


              SEGMENT $MAIN$          040000 040015

ERU$/SYS72-8
B                         $(0)    040000 040015

SYS$*RLIB$. LEVEL 72-8
END MAP


@XQT      ABS                                                          .      ***


@FIN                                                                   .      ***
```

EUROPEAN EDUCATION AND TRAINING CENTER

SECTION 6

MARGIN VOLTAGE ADJUSTMENTS

CONTENTS LIST

		Page No.
6.1	DC VOLTAGE ADJUSTMENT	1
6.2	VOLTAGE MARGIN TESTING	1
6.3	VOLTAGE DISTRIBUTION	2

ILLUSTRATIONS

6.1	POWER SUPPLIES	1

6 MARGIN VOLTAGE ADJUSTMENTS

6.1 DC VOLTAGE ADJUSTMENT

DC voltages should be adjusted with a recently calibrated Weston Meter (± 1%) on the following test pins.

-3A	3A44–4
-3B	1D44–4
-3C	4C44–4
-6	2D44–52
-12	2C44–46
+12	At supply
+6	3B64–28

The readings on the percentage deviation meter should then be taken and noted as NOMINAL readings alongside the meter.

e.g. -3A NOMINAL = + 1%

6.2 VOLTAGE MARGIN TESTING

At installation, the margin limits should be taken and again noted as percentage deviation from NOMINAL, so that any deterioration at a later date may be noticed.

On later CPU's above S/N 132, a total margin range of 15% should be possible on each supply except for -3A where it will probably be a little less than this. Acceptable DC voltage margins for Unit Processors are ± 6% except -3A Volt which is + 6% -4%. For Multi Processor it is normally better to maintain margins as ± 8% and + 8%, - 6% for -3A.

For faults of a marginal or intermittent nature, margin testing is very valuable, each voltage should be swung in turn and any discrepancies rectified even if superficially they do not resemble the fault in hand. The voltages may be swung if necessary up to ± 10% without causing damage to any circuitry. (With the exception of -3A, where more care is advisable).

Margin testing will help in detecting bad logic levels, noisy components, marginal timing and noisy power supplies.

Although generally the Command Arithmetic and ICR tests are used for voltage testing, obscure faults might be intensified by running as many DVR tests as possible, or even Exec itself during customer running (a range from + 5% to – 4% might be tried).

For the latter the following points should be realised:

1) The margins are variable depending on the runs in operation.

2) Programming assistance is invariably required if a fault occurs.

3) A failure might corrupt vital drum data, necessitating a lengthy tape boot.

For these reasons, margins during Exec running should only be used when everything else fails.

VOLTAGE DISTRIBUTION

-3A	A03A	A03B	
-3B	A01 (all)	A03C	A03D
-3C	A02 (all)	A04 (all)	
-6	A03A	A03B	
-12	All cards		
+ 6	A03A		
+ 12	Maintenance Panel Console		

Looking from the rear of the processor the supplies are located as shown below:

Common 100V A08	-12 A04	-6 A02
	+ 12 A05	-3A A15
-3C A09	+ 6 A06	-3B A03

Figure 6.1: Power Supplies

Each supply has two potentiometers (V–ADJ) to adjust the voltage output and the upper limit which will determine the upper voltage limit should the regulator circuit fail. The latter will have to be adjusted to allow the voltage to be sufficiently high for marginal testing.

An old PC Board could be modified so that pin 4 (-3V) pin 52 (-6V) pin 64 (-12V) and pin 28 (+6V) are brought to test points on the card and all other connections removed.

This can then be inserted into a spare location for voltage adjustment. It would be safer than measuring on the back panel itself.

SECTION 7

CENTRAL PROCESSOR

CONTENTS LIST

		Page No.
7.1	SHOCK TESTING	1
7.2	UNITS POWER ON/OFF	1
7.3	ELECTRICAL NOISE	1
7.4	CLOCK DISTRIBUTION AND TIMING	1
7.5	CLOCK STOP	2
7.6	BREAKPOINT	4
7.7	FIXED EXAMPLES OF FLOATING POINT MULTIPLY	4
7.8	PROGRAM FOR CTMC WIRED IN BACK TO BACK OPERATION	7

7 CENTRAL PROCESSOR

7.1 SHOCK TESTING

The 1108/06 is given a very thorough shock testing on the test floor, the back panel wiring is also blown out to remove wire snips, cold flows etc. Ground connections, power supply connections etc. are also checked for tightness.

For efficient running, the 1108/06 should not be shock sensitive.

If intermittent problems occur, which voltage margins will not resolve, then shock testing may be employed. Try it whilst running C/A and ICR test; if an error occurs, set jump 4 and repeat the failing test.

The main causes of failure are splayed female pins on the cards, but back panel pins may be pushed out, components damaged or cards shorting between printed circuit and adjacent component.

Remember that cables may also develop bad contacts and should be shock tested.

Shock testing can be very valuable, but there are four considerations:

1) Common sense must be used, any CPU can be damaged with perserverance.
2) Shock only smaller portions and then verify operation of CPU. If necessary, try EXEC between all steps, because faults may be introduced which are not detected by DVR.
3) Despite the above, faults may be induced. Allow sufficient time to clear any faults which occur.
4) To test as much circuitry in the CPU as possible, also run Memory Test and Subsystem Tests.

7.2 UNITS POWER ON/OFF

The electronic equipment shall preferably be left powered on at all times. Power on/off puts extra strain on the units and components, and causes additional failures.

If the units are switched off, allow sufficient time to cool down to avoid overheating. Airconditioning should not be removed until 1 hour after power off.

If a system which is normally left continuously on, has to be powered down, schedule, if possible, sufficient test time after power up, to rectify any faults caused by the existence of weak components.

7.3 ELECTRICAL NOISE

Intermittent failures can be caused by noise spikes on the DC voltage supplies. This can be checked and shall peak to peak not exceed 10% of the DC voltage. A good power supply has less than 5% noise peak to peak. Make sure that the scope is only grounded to logic ground.

Cause for noise on DC power may be bad power, faulty regulation or poor systems grounding. For system ground checks, see PM summary FEC 2007. No deviations from the grounding method required for large scale systems can be tolerated. Good grounding and good main power is the base of reliable systems operation and an absolute requirement for troubleshooting highly intermittent faults.

7.4 CLOCK DISTRIBUTION AND TIMING

Clock adjustment is contained in the 1108 maintenance handbook. That is:

1) Clock pulse measurements are made at the backboard pins on A03J06D.

   01 Pin 20 TP 8
   02 Pin 28 TP 10
   03 Pin 36 TP 12
   04 Pin 48 TP 14

2) Use output at A03J41D–36 test point 11 (continuous 01) as an external synch. for the scope.

3) The clock cycle time is fixed by an oscillator with a cycle time of 125 ± 0.1ns. Adjust the width of each of the clock phases to 42.5 ± 0.25ns. measured at –2.2 volts. Adjust the delays between leading edges as follows:

   Ø1 and Ø2 31.25 ± 0.25ns.
   Ø1 and Ø3 62.50 ± 0.25ns.
   Ø1 and Ø4 93.75 ± 0.25ns.

   Note: Check scope calibration at 3D4TP14 which is the output of the CPU master clock. This is most important.

4) For 1106 the clock cycle time is 167ns. The phase separation is,

   Ø1 and Ø2 41.75 ± 0.50ns.
   Ø1 and Ø3 83.50 ± 0.50ns.
   Ø1 and Ø4 125.25 ± 0.50ns.

5) The clock is not very critical, and margins of ± 3–4ns. should be possible.

6) It is worthwhile checking the distribution outputs for good pulses, especially if there are poor machine margins. The following is a list of the outputs:

Fast 01	Fast 02	Fast 03	Fast 04
A04 A52 TP 2	A04 A52 TP 8	A03 D06 TP 2	A04 A52 TP 10
A04 A52 TP 4	A04 A55 TP 11	A04 A52 TP 12	A03 D06 TP 4
A02 A58 TP 2	A02 A58 TP 10	A02 A59 TP 2	A02 A59 TP 10
A02 A58 TP 4	A02 A58 TP 12	A02 A59 TP 4	A02 A59 TP 14
A02 A58 TP 8	A02 A59 TP 12	A04 A54 TP 9	A04 A54 TP 11
A04 A52 TP 14	A04 A54 TP 7	A04 A54 TP 10	A04 A54 TP 13
A04 A55 TP 9	A04 A54 TP 8	A01 D48 TP 12	A01 D53 TP 12
A02 A58 TP 14		A01 D48 TP 14	A01 D53 TP 14
A02 A59 TP 8			

Slow 01	Slow 02	Slow 03	Slow 04
A04 B30 TP 2	A04 A55 TP 15	A04 B30 TP 12	A04 B30 TP 8
A04 B30 TP 4	A04 A55 TP 13	A04 B30 TP 14	A04 B30 TP 10
A04 C57 TP 7	A04 C57 TP 9	A04 C57 TP 13	A04 C57 TP 10
A04 C57 TP 8	A04 C57 TP 11	A04 C57 TP 15	A04 C66 TP 10
A02 A69 TP 2	A04 A69 TP 4	A02 A69 TP 10	A02 A69 TP 12
A01 D53 TP 2	A02 A69 TP 8	A01 D53 TP 8	A02 A69 TP 14
A01 C61 TP 8	A01 D53 TP 4	A03 C72 TP 7	A01 D53 TP 10
	A04 C42 TP 5		

7.5 CLOCK STOP

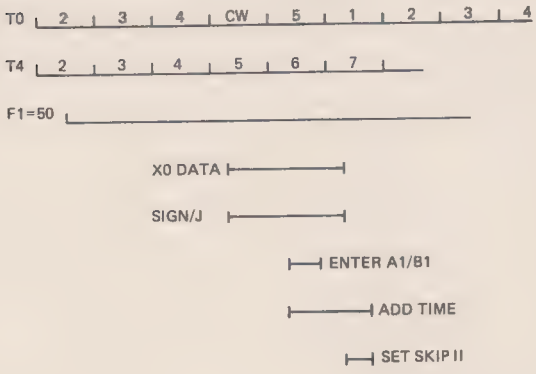
This has already been mentioned under ICR troubleshooting hints.

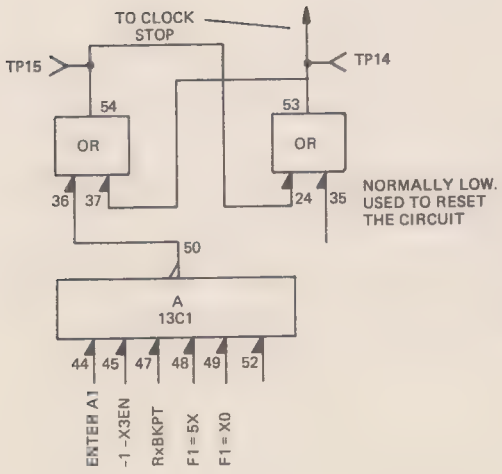
Reference to Main Control Logics, page 138, will show a high on 3D4 pin 20 which will stop the clock.

This can be used for a variety of purposes, one example is a trap shown below, which was used on a fault where a test zero of a zero location was very occasionally failing during EXEC running.

Normally the data will be lost by the time that the error is detected, and this trap was generated to stop the clock, so that picking of a bit should be checked for in X0, should Skip II be set.

Note that a flip-flop circuit is used to ensure a long enough "high" to completely block the clock circuit.





ANY SPARE PACKAGE WITH SUITABLE ELEMENTS MAY BE USED IN ANY SPARE LOCATION. 13C1, WHICH IS AN INVERTER PACKAGE, WAS USED IN THIS INSTANCE

7.6 BREAKPOINT

Refer to Main Control Logics, page 138.

The uses of the breakpoint register are:

a) To stop the computer, when the contents of P or R are equal to the breakpoint register, provided that BRKPT SEL or BRKPT SEL and R SEL, respectively, are set.

b) If BRKPT SEL is not set, the computer will not stop, but a pulse will be produced at equality from 4B74TP2. The main use of this pulse is as a scope sync.

For example: Some faults will fail on margins, but give the correct result when phase stepping through. A method to locate the fault is:

1) Assume a floating add is failing.

2) Obtain from the test program or otherwise, a pair of failing operands.

3) Write and button-in a program such as:

```
          FO j  a  b  hi    u
1000      10 0  1           1100  (u)  A1
1001      76 0  1           1102  (A1) +u A, A+1
1002      74 4  1           1000  Jump to u if JK=1
1003      52 0  1           1103  (u) = (A)
1004      74 5  1           1000  Halt if stop 1
1005      52 0  2           1104  (u) = (A+1)
1006      74 5  2           1000  Halt if stop 2
1007      74 4  0           1000  Jump to u
```

A (1100) = 606.340000000
u (1102) = 603.240000000
Result: (1103) = 603.174000001
Residue: (1104) = 636.077777777

4) With the aid of stop 1 or 2, adjust the margins so that the CPU will fail 50% of the time.

5) With 1001 in the Breakpoint reg. set Jump Key 1 and sync. the scope off 4B74 TP 2 (Breakpoint Sync.).

6) Look along the registers for a failing pulse.

7) With expanded sweep and a block diagram this may be traced back to its failing point.

7.7 FIXED EXAMPLES OF FLOATING POINT MULTIPLY

There is an example of floating point multiply in the 1108A maintenance handbook but this is somewhat lengthy and an example such as that given here might be more convenient for arithmetic faults.

This type of example is primarily used for phase stepping through an instruction.

f = 76 j 12 A - Q0, U - A0
A+1 - Q2, U+1 - A2

X0 3.4 X1 3.4
(A) = 201165432100, (A+1) = 100000000000
(U) = 177265045404, (U+1) = 500000000000
X0 5.3 X1 5.3

Note: all registers shown will be read $2^{35} - 2^0$, B0 2^{37} and A2 2^{36} and 2^{37} will be shown separately.

Cycle 1 Pass 2

Phase		
02	after multiply control 3 sets multiply control 4 sets	
03	Enter A1B1, A3B3 = neg. 0	
04	$Q1_{R4}$ - Q0, $Q3_{R4}$ - Q2	
01	Enter A0B0, A2B2 = 0. . . .0	
02	Q0 - Q1, Q2 - Q3	
03	Enter A1B1, A3B3 = neg. 0	
04	$Q1_{R4}$ - Q0, $Q3_{R4}$ - Q2	
	next 7 cycles the same	
	9th cycle:	
01	Enter A0B0, A2B2	
	A0 000013715641	
	B0 000002504440	B0 Bit 37 = 1
	A2 740. 0	A2 Bit 36 = 0, Bit 37 = 1
	B2 240. 0	
02	Q0 - Q1, Q2 - Q3	
03	Enter A1B1, A3B3	
	A1 777761273037	
	B1 777765273137	
	A3 327. 7	
	B3 327. 7	
01	Enter A0B0, A2B2	
	A0 000001574652	
	B0 000000524212	B0 Bit 37 = 0
	A2 3360. 0	A2 Bit 36 = 0, Bit 37 = 0
	B2 1120. 0	
03	Enter A1B1, A3B3	
	A1 777777053501	
	B1 777777253525	
	A3 75537. 7	
	B3 75537. 7	

EUROPEAN EDUCATION AND TRAINING CENTER

Phase		
01	Enter A0B0, A2B2	
	A0 777762501240	
	B0 377762525252	
	A2 66207. 7	
	B2 66657. 7	
03	Enter A1B1, A3B3	
	A1 777750024012	
	B1 777755276537	
	A3 310117. 7	
	B3 73337. 7	
01	Enter A0B0, A2B2	
	A0 777761440025	
	B0 377761566477	B0 Bit 37 = 1
	A2 1402237. . . . 7	A2 Bit 37 = 1
	B2 3557. 7	A2 Bit 36 = 0
03	Enter A1B1, A3B3	
	A1 000047750556	
	B1 000047355556	
	A3 456733000000	
	B3 546733000000	
01	Enter A0B0, A2B2	
	A0 777764125505	
	B0 377764525555	B0 Bit 37 = 1
	A2 507144577777	A2 Bit 37 = 1
	B2 267355577777	A2 Bit 36 = 1
03	Enter A1B1, A3B3	
	A1 777610512150	
	B1 777611652372	
	A3 765420037777	
	B3 577566677777	
01	Enter A0B0, A2B2	
	A0 777773027347	
	B0 377763167277	B0 Bit 37 = 0
	A2 175200447777	A2 Bit 37 = 0
	B2 277673777777	A2 Bit 36 = 0

Phase		
03	Enter A1B1, A3B3	
	A1 777727443542	
	B1 777706752572	
	A3 777202111777	
	B3 777737.....7	
01	Enter A0B0, A2B2	
	A0 000010706030	
	B0 000010615211	B0 Bit 37 = 0
	A2 300117355400	A2 Bit 37 = 0
	B2 300117355400	A2 Bit 36 = 1
03	Enter A1B1, A3B3	
	A1 000031334412	
	B1 000021030402	
	A3 720064315500	
	B3 500044104400	
01	Clear A0	
04	Enter A0B0, A2B2	
	A0 777767253405	
	B0 000010615211	B0 Bit 37 = 0
	A2 477660422374	A2 Bit 37 = 1
	B2 300117355400	A2 Bit 36 = 0
01	Clear B0B2	
03	Enter A1B1, A3B3	
	A1 000075561776	
	B1 000010520372	
	A3 760023673300	
	B3 260023673300	
01	Clear A0A2	
03	A0A2 0 – – 0	
04	Clear Ext. Seq.	
02	Set overflow	
04	Clear overflow	
02	X7 – A2 = 217754104500	
03	X4 – A0 = 200354321012	

7.8 PROGRAM FOR CTMC WIRED IN BACK TO BACK OPERATION

The program which follows can be run through a CTMC wired for back to back operation.

An addressing fault in storing the ESI data word can cause confusing symptoms in customer operation or

engineering tests.

The program can be enhanced by address tagging the O/P Buffer, but remember that the halves or quarters of the O/P words should be differently tagged also.

P	f	j	a	b	hi	u	
300	10	00	00			312	OP–ACW→A0
301	10	00	01			313	IP–ACW→A1
302	01	00	00			ZZZZZ	A0→O/P Z–Z = Base + CLT Address
303	01	00	01			ZZZZZ+1	A1→I/P
304	75	14	00			00000	AACI (allow all EI)
305	74	07	00			306	AAIJ (allow all I/O Intrpt.+Jump)
306	75	10	XX			314	LFC XX = CTMC channel
307	75	00	XX				LIC XX
310	75	05	XX				LOCM XX
311	74	05	01			311	HKJ Stop with Key 1
312	00	00	20	00	06	00	OP–ACW
313	00	00	20	00	10	00	IP–ACW
314	00	00	01	00	03	15	Funct.–ACW
315						YYY	CLT Function Y–Y = Function
600							
		Output Buffer					
620							
1000							
		Input Buffer					
1020							
225	74	05	02			300	HKJ Stop Key 2 OP–Monitor

EUROPEAN EDUCATION AND TRAINING CENTER

SECTION 8

MEMORY

CONTENTS LIST

		Page No.
8.1	7005 MEMORY TROUBLESHOOTING	1
8.2	FAULT ISOLATION	1
8.3	MEMORY MAINTENANCE PANEL	1
8.4	CPU MAINTENANCE PANEL	2
8.5	MEMORY TEST	2
8.6	CUSTOMER PROGRAM	3
8.7	INTERMITTENT FAULTS	3
8.8	MEMORY FAULTS WITHOUT PARITY FAULTS	3
8.9	ADDRESSING PROBLEMS	3
8.10	7005 MEMORY TROUBLESHOOTING BASED ON SYMPTOMS ON MAINTENANCE PANEL	4
8.11	MARGIN TEST ON 7005–07/08 MEMORY	12

TABLES

8.1 – 8.9	7005 MEMORY TROUBLESHOOTING INFORMATION	6–11

ILLUSTRATIONS

8.1	MODULE LOCATIONS	5
8.2	MEMORY ADDRESS SELECTION	5
8.3	UNIT PROCESSOR SYSTEMS WITHOUT INTERLEAVING FEATURE (7005)	13
8.4	VERTICAL MODE WITH INTERLEAVING FEATURE (7013)	14
8.5	UNIT OR MULTIPROCESSOR SYSTEM WITH INTERLEAVING FEATURE (F 1053–00)	15
8.6	VERTICAL MODE WITH INTERLEAVING FEATURE (7013)	16
8.7	UNIT OR VERTICAL MODE WITHOUT INTERLEAVING FEATURE (7013)	17

8 MEMORY

8.1 7005 MEMORY TROUBLESHOOTING

This outline is primarily intended as a guide for engineers who have completed theoretical training on the memory but who have not yet had much experience practically. Although it is referenced to the 7005 memory on the 1108, the general philosophy should be the same on most other computers and memories.

There are several types of faults which may occur on memories:

1) Parity faults.
2) Dropping or picking up bits without parity faults.
3) Addressing faults.
4) Timing faults.

These faults might be further divided into two categories: Solid and Intermittent. Each requires a different approach in troubleshooting.

No attempt is being made here to provide all faults and cures. However, it is hoped that having studied it, the new engineer will be able to cope better with them. Most of the considerations are based on past faults and problems, encountered in the field, thereby exposing the new engineer to some past experience.

8.2 FAULT ISOLATION

Basically there are four general approaches possible for fault isolation.

1) From the memory maintenance panel.
2) From the CPU maintenance panel.
3) Using DVR. (A diagnostic memory test is released to the field).
4) Using customer runs.

Each one of these has relative advantages and disadvantages depending on the fault, the time available, and the personal preference of the engineer.

8.3 MEMORY MAINTENANCE PANEL

If the fault occurs in this way, it provides a convenient means of isolation, especially if the customer can continue operation with one cabinet less.

At least one failing address should be known before attempting to use the memory maintenance panel. In this way, the failing bit or bits can be determined by writing and then reading.

By changing the Y selects, X selects and octant selects of this address, one at a time, the suspect area can often be narrowed down to one element. It should be noted that X selects and octant selects usually affect a whole half word, Y selects affect only one bit, and a faulty sense amplifier may affect one or all octants, depending on the faulty element. More details are provided in the chart, (Appendix A). It is advisable to study the chart and to consult the logics for any points which might not be clear. In most cases, where the fault is solid, this approach should be sufficient. It must be remembered, though, that such a write up cannot cover all possibilities and must be used with a certain amount of discretion.

8.4 CPU MAINTENANCE PANEL

The advantage of this system is that the first failing address can be found rapidly. A sound approach to troubleshooting the memory is to record the failing address and data at the time of failure. It is worth mentioning at this time that the Stop On Parity switch on the CPU will not help here. The address obtained in this way is the one following the failing address but is not necessarily the consecutive one. (Remember vertical addressing and also the fact that the instructions may jump and that the u portions are rarely sequential). In order to be sure of the address at the time of failure, the memory itself must be in "Stop On Error". This might be inconvenient during customer operation as it inhibits the parity fault indication on the CPU and console, giving a hang-up of the CPU and a parity fault light on the memory. The failing address can be found quickly by enabling "Stop On Error" on the failing memory and reading it using the procedure in Appendix.

When the failing address is reached, stop the processor, note the data and H address, clear the parity fault and start the processor. In most cases following customer operation, only one failing address will be found.

At this point, the failing address can be written and read either from the memory or CPU maintenance panel to find the failing addresses. Once this is established, the memory should be as good as fixed.

8.5 MEMORY TEST

Two memory tests are available on the later revisions of master: one which loads on its own and another which runs concurrently with other tests under DVR EXEC.

Learn how to operate both of these as soon as possible after arriving on site and question experienced engineers about their operation. Learn how to bootstrap into any memory and restart the tests following program "blow-ups". Learn how to use the jump and stop switches on the console. Find out about the addressing scheme on your site. There are two possibilities depending on whether the system is a unit or multi-processor.

These are:

1) Unit Addressing.

2) Vertical Addressing.

(Horizontal addressing is not in use in Europe). Look these up in the manual and consider their affect on the memory test, particularly with respect to the core being tested.

The core test can be run with "Stop On Parity" on the memory as was mentioned before. Failing bits and addresses can be observed on the memory maintenance panel. Note that the addresses in the type-outs are not the addresses observed on the memory because of unit or vertical addressing.

Whether or not the test is used for troubleshooting is usually a matter of personal choice. In cases where a fault cannot be reproduced "off-time", the test might well fail and should be tried out. This could be particularly true in the case of shock-sensitive elements or marginal components. Following intermittent memory faults, investigate these possibilities by checking the supply and sense bias margins to make sure they are up to limits achieved the last PM; shock sensitivity can also be checked out. Intermittent problems are discussed more fully below.

8.6 CUSTOMER PROGRAM

This is definitely the least desirable approach to troubleshooting and should only be used where all else fails to turn up the fault. It is difficult to locate the failing bit in this way but occasionally, with the help of system programmers, it is possible. Remember again that the address on the memory maintenance panel must be converted back from unit or vertical addressing, to be useful to the programmer. Another fact to bear in mind is that the trouble might be external, as discussed under intermittent faults.

8.7 INTERMITTENT FAULTS

Except for their frequency, the intermittents are essentially the same as solid faults. Depending on their cause, however, they may or may not be logically associated. Faults caused by shock sensitivity or marginal components, for example, would normally show a consistency which should lead to a faulty element. They can often be detected by supply or sense bias margin testing or shock testing. The memory should run on margins obtained during the last past months and should not fail shock testing.

In cases where a memory gives frequent faults, all pertinent data such as MDR contents, H register contents, and time of day should be recorded. An effort should be made to find out which bit is failing and consistencies between addresses should be checked out.

At the same time, all other possibilities should be checked out, are the power supplies noisy or mis-adjusted? Are the power lines noisy? Is there ground noise? Is there a possible source of radiated noise? Are there any other problems on the system that might be associated to the memory problem? Is the air-conditioning stable? These are only a few points to bear in mind while tracing down intermittent faults.

8.8 MEMORY FAULTS WITHOUT PARITY FAULTS

Dropping or picking up bits —

Assuming the fault occurs during the memory test, check for the following:

Which bit is failing? See type-out. Does it show up on the memory? Is the bit being written? With which tests? (If only on address tagging, see address faults). From all memories? It could be in the processor interface. Does the fault occur when a program is loaded into this memory? If not, the processor is most likely at fault. Does the fault occur when the memory is operated from the CPU maintenance panel? Remember that all "1s" and all "0s" functions can be tried to check the path from memory to F0. (Ignore the instructions fault).

A common symptom of this fault is inability to bootstrap, the "boot" hang-up following the first block on an instruction fault. Evidently this type of fault could come from other areas as well, such as I/O, servo control, CPU, MMA, SPI, etc. and must be pinned down before troubleshooting starts. A useful procedure in such a case is to bootstrap on all "1s" and all "0s" tape. In this way it can be determined whether the fault is in reading or writing and where the fault is localized.

8.9 ADDRESSING PROBLEMS

This type may be more difficult to pin down and might even be confused with the dropping or picking up of bits. Consider the affect of dropping an address bit on the memory test. The all "1s", all "0s" test would most likely work. Address tagging, however, would appear to be picking up a bit. Compliment address tagging would appear to drop the same bit. This is because the test, upon reaching the failing address bit, would go back to address Ø with the data for a higher address. On the read following this,

address Ø would contain a tag for the higher address and the compare would fail, giving an error type-out indicating bit pick-up or dropping. (This fact holds true for all devices which require addressing).

Again the problem becomes more difficult if it is intermittent. In many cases, it will result in such things as EXEC blow-up and probably will not showup in off-line operation. Consider the affect on memory operation of a faulty G to H transfer (picking up or dropping bits).

If error shows on engineering test, then breakpointing on error exit and checking the R registers against G and H registers may help.

Parity errors without missing bits are most frequently caused by the following:

1) Parity bit bad.
2) Parity errors on partial writers.
3) Faulty parity tree.

The first case should be relatively easy to detect and is handled as a bad data bit.

Parity errors exclusively on partial writes can occur and are somewhat difficult to track down, as the data is never staticized in the MDR. By synchronizing on parity error and looking at the sense output for each bit, the faulty element might be found. This technique of comparing signals is often useful in trouble shooting but must be used with care. In many cases, all the conditions may not be the same on the elements being compared and the difference can be normal.

7005 MEMORY TROUBLESHOOTING BASED ON SYMPTOMS ON MAINTENANCE PANEL

The following chart and tables have been developed as an aid for quickly isolating many types of failures down to a PC board. To use the chart, simply locate the failing bit symptoms in column 1 and correlate with the failing address symptom located adjacent to it in column 2. The PC cards that control the given situation are then found immediately to the right in column 3, or if there are multiple possibilities you are referred to an appropriate table.

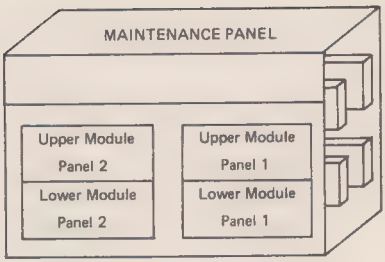


Fig. 8.1: Module Locations

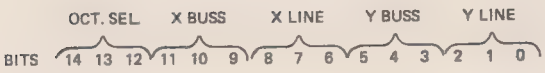


Fig. 8.2: Memory Address Selection

Error Conditions in MDR	Address in MAR	See Table No. or PC Card Loc.	Logic Pages PX 3079
WHOLE WORD IS FAILING	On one or more address Sel.	Table 8.2	7--11
LOWER HALF WORD IS FAILING 494: BITS 0--14 1108: BITS 0--17	Depending on X Buss Sel. together with Octant Sel.	Table 8.3 (PC Boards in Stack 1)	62--65
	Depending on X Buss	Read Write 1C10 1C20	24 , 25
	Depending on X Line Sel. together with Octant Sel.	Table 8.4	16--23
	Depending on Octant Sel.	Table 8.5	27 30
UPPER HALF WORD IS FAILING 494: BITS 15--29 1108: BITS 18--35	Depending on X Buss Sel. together with Octant Sel.	Table 8.3 (PC Boards in Stack 2)	62 65
	Depending on X Buss	Read Write 1D10 1D20	24 , 25
	Depending on X Line Sel. together with Octant Sel.	Table 8.6	16--23
	Depending on Octant Sel.	Table 8.7	27--30
494: BITS 0--7 & PARITY 1 OR 15--22 & PARITY 2 OR BOTH ARE FAILING 1108: BITS 0--7 & PARITY 1 OR 18--25 & PARITY 2 OR BOTH ARE FAILING	Depending on Y Buss	Read Write 1B54 1B56	14 , 15
	Depending on Y Line	Read Write Evn. S. 1B52 1B56 Odd S. 1B54	12 , 13
494: BITS 8--14 OR BITS 23--29 OR BOTH ARE FAILING 1108: BITS 8--17 OR BITS 26--35 OR BOTH ARE FAILING	Depending on Y Buss	Read Write 2C04 2B10	14 , 15
	Depending on Y Line	Read Write Evn. S. 2D04 2D04 Odd S. 2C04	12 , 13
ONE BIT FAILING	Sense Amplifier (Any address)	Table 8.8	33--42
	Y Matrix Select (Any address)	Table 8.9	31 , 32

Table 8.1

Area Sel.	Oct. Sel. 0–7	
	Read	Write
Y LINE	2B02	2B06
Y BUSS	2B04	2B08
X LINE	1B02	1B08
X BUSS	1B04	1B10
OCT. SEL.	1B06	1B12

Table 8.2

Octant Sel	X Buss * 0 & 1	X Buss * 2 & 3	X Buss * 4 & 5	X Buss * 6 & 7
0	1	2	3	4
1	5	6	7	8
2	9	10	11	12
3	13	14	15	16
4	17	18	19	20
2	21	22	23	24
6	25	26	27	28
7	29	30	31	32

* X Sel. Trans. Module Cards are placed on the Stack. The shelf begins with Module Ø Lowest.

Table 8.3

X Line Sel.		Group 1 Oct Sel. 0 & 1	Group 2 Oct Sel. 2 & 3	Group 3 Oct Sel. 4 & 5	Group 4 Oct Sel. 6 & 7
0	READ	1C02	1C04	1C06	1C08
	WRITE	1C12	1C14	1C16	1C18
1	READ	1C02	1C04	1C06	1C08
	WRITE	1C12	1C14	1C16	1C18
2	READ	1C02	1C04	1C06	1C08
	WRITE	1C12	1C14	1C16	1C18
3	READ	1C02	1C04	1C06	1C08
	WRITE	1C12	1C14	1C16	1C18
4	READ	1C02	1C04	1C06	1C08
	WRITE	1C12	1C14	1C16	1C18
5	READ	1C02	1C04	1C06	1C08
	WRITE	1C12	1C14	1C16	1C18
6	READ	1C02	1C04	1C06	1C08
	WRITE	1C12	1C14	1C16	1C18
7	READ	1C02	1C04	1C06	1C08
	WRITE	1C12	1C14	1C16	1C18

Table 8.4

Oct Sel.	Read	Write
0	1B38	1B38
1	1B36	1B38
2	1B36	1B38
3	1B36	1B38
4	1B42	1B42
5	1B40	1B42
6	1B40	1B42
7	1B40	1B42

Table 8.5

X Line Sel.		Group 1 Oct Sel. 0 & 1	Group 2 Oct Sel. 2 & 3	Group 3 Oct Sel. 4 & 5	Group 4 Oct Sel. 6 & 7
0	READ	1D02	1D04	1D06	1D08
	WRITE	1D12	1D14	1D16	1D18
1	READ	1D02	1D04	1D06	1D08
	WRITE	1D12	1D14	1D16	1D18
2	READ	1D02	1D04	1D06	1D08
	WRITE	1D12	1D14	1D16	1D18
3	READ	1D02	1D04	1D06	1D08
	WRITE	1D12	1D14	1D16	1D18
4	READ	1D02	1D04	1D06	1D08
	WRITE	1D12	1D14	1D16	1D18
5	READ	1D02	1D04	1D06	1D08
	WRITE	1D12	1D14	1D16	1D18
6	READ	1D02	1D04	1D06	1D08
	WRITE	1D12	1D14	1D16	1D18
7	READ	1D02	1D04	1D06	1D08
	WRITE	1D12	1D14	1D16	1D18

Table 8.6

Oct Sel.	Read	Write
0	1B46	1B46
1	1B44	1B46
2	1B44	1B46
3	1B44	1B46
4	1B50	1B50
5	1B48	1B50
6	1B48	1B50
7	1B48	1B50

Table 8.7

Bit No.	494 Sense Amplif. Locat.	1108 Sense Amplif. Locat.	Bit No.	494 Sense Amplif. Locat.	1108 Sense Amplif. Locat.
0	1A42	1A42	19	2A06	1A56
1	1A44	1A44	20	2A08	2A02
2	1A46	1A46	21	2A22	2A04
3	1A48	1A48	22	2A24	2A06
4	1A50	1A50	23	2A26	2A08
5	1A52	1A52	24	2A28	2A22
6	2A10	2A10	25	2A30	2A24
7	2A12	2A12	26	2A32	2A26
8	2A14	2A14	27	2A46	2A28
9	2A16	2A16	28	2A48	2A30
10	2A18	2A18	29	2A50	2A32
11	2A20	2A20	30	– –	2A46
12	2A34	2A34	31	– –	2A48
13	2A36	2A36	32	– –	2A50
14	2A38	2A38	33	– –	2A52
15	1A54	2A40	34	– –	2A54
16	1A56	2A42	35	– –	2A56
17	2A02	2A44	P1	1A38	1A38
18	2A04	1A54	P2	1A40	1A40

Table 8.8

Bit No.	494				1108			
	Read		Write		Read		Write	
	Card Loc.	Plug No.	Card Loc.	Plug No.	Card Loc.	Plug No.	Card Loc.	Plug No.
0	1C26	P04	1C28	P04	1C26	P04	1C28	P04
1	1C30	P06	1C32	P06	1C30	P06	1C32	P06
2	1C34	P08	1C36	P08	1C34	P08	1C36	P08
3	1C38	P10	1C40	P10	1C38	P10	1C40	P10
4	1C42	P09	1C44	P09	1C42	P09	1C44	P09
5	1C46	P12	1C48	P12	1C46	P12	1C48	P12
6	1C50	P11	1C52	P11	1C50	P11	1C52	P11
7	1C54	P14	1C56	P14	1C54	P14	1C56	P14
8	2C06	P13	2C08	P13	2C06	P13	2C08	P13
9	2C10	P16	2C12	P16	2C10	P16	2C12	P16
10	2C14	P15	2C16	P15	2C14	P15	2C16	P15
11	2C18	P18	2C20	P18	2C18	P18	2C20	P18
12	2C22	P17	2C24	P17	2C22	P17	2C24	P17
13	2C26	P19	2C28	P19	2C26	P19	2C28	P19
14	2C30	P21	2C32	P21	2C30	P21	2C32	P21
15	1D26	P34	1D28	P34	2C34	P20	2C36	P20
16	1D30	P36	1D32	P36	2C38	P22	2C40	P22
17	1D34	P38	1D36	P38	2C42	P24	2C44	P24
18	1D38	P40	1D40	P40	1D26	P34	1D28	P34
19	1D42	P39	1D44	P39	1D30	P36	1D32	P36
20	1D46	P42	1D48	P42	1D34	P38	1D36	P38
21	1D50	P41	1D52	P41	1D38	P40	1D40	P40
22	1D54	P44	1D56	P44	1D42	P39	1D42	P39
23	2D06	P43	2D08	P43	1D46	P42	1D48	P42
24	2D10	P46	2D12	P46	1D50	P41	1D52	P41

Table 8.9

Bit No.	494				1108			
	Read		Write		Read		Write	
	Card Loc.	Plug No.	Card Loc.	Plug No.	Card Loc.	Plug No.	Card Loc.	Plug No.
25	2D14	P45	2D16	P45	1D54	P44	1D56	P44
26	2D18	P48	2D20	P48	2D06	P43	2D08	P43
27	2D22	P47	2D24	P47	2D10	P46	2D12	P46
28	2D26	P49	2D28	P49	2D14	P45	2D16	P45
29	2D30	P51	2D32	P51	2D18	P48	2D20	P48
30	– –	– –	– –	– –	2D22	P47	2D24	P47
31	– –	– –	– –	– –	2D26	P49	2D28	P49
32	– –	– –	– –	– –	2D30	P51	2D32	P51
33	– –	– –	– –	– –	2D34	P50	2D36	P50
34	– –	– –	– –	– –	2D38	P52	2D40	P52
35	– –	– –	– –	– –	2D42	P54	2D44	P54
Parit. 1	1C22	P02	1C24	P02	1C22	P02	1C24	P02
Parit. 2	1D22	P32	1D24	P32	1D22	P32	1D44	P32

Table 8.9 (Cont.)

8.11 MARGIN TEST ON 7005–07/08 MEMORY

A more precise way of testing the sense bias margin range on a 7005–07/08 memory is, to run the test for each 16K of storage separate.

To check the upper limit the margin test card (p/n 410 1828) can be modified by shorting the 5 ohm resistor. In addition, the –15 Var power supply can be increased by +15 percent. This will not affect the Read/Write Drivers but allow testing up to –15 to 16 volts of the upper limit.

A typical margin range on a new 7005–08 memory (U1108) will be around 9 volts (between 6 and 15 volts), the margin range on a 7005–07 memory (U494) will be around 6 volts. To achieve equal margin ranges for both 16K of storage in one 32K module, the strobe timing can be adjusted. Moving the strobe to an earlier tap allows the margin range to go to a higher upper limit.

Note: Do not move the strobe to a later tap on a 7005–08 memory (U1108) as this can give a too long access time.

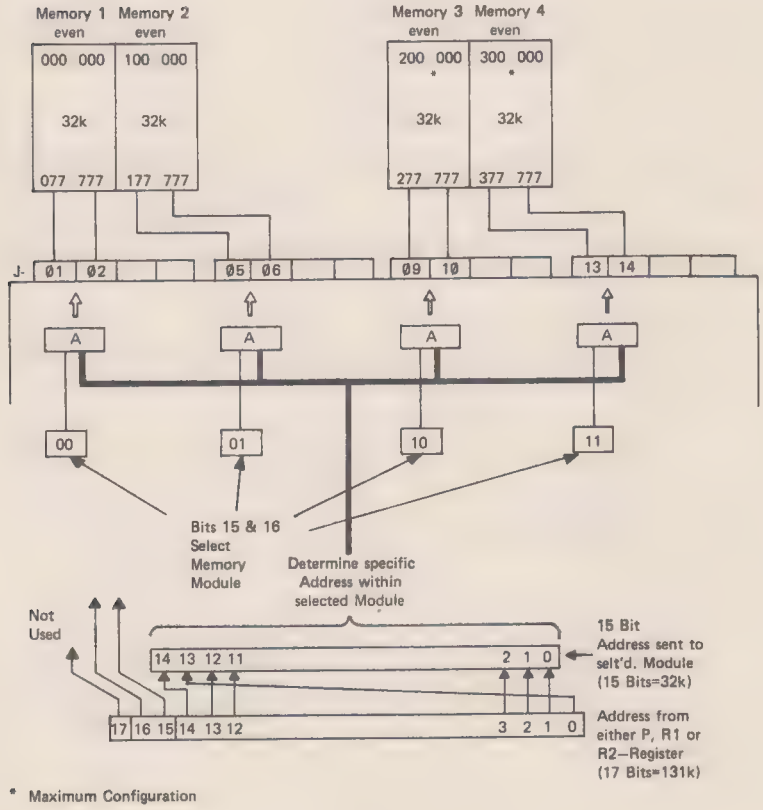


Fig. 8.3: Unit Processor Systems Without Interleaving Feature (7005)

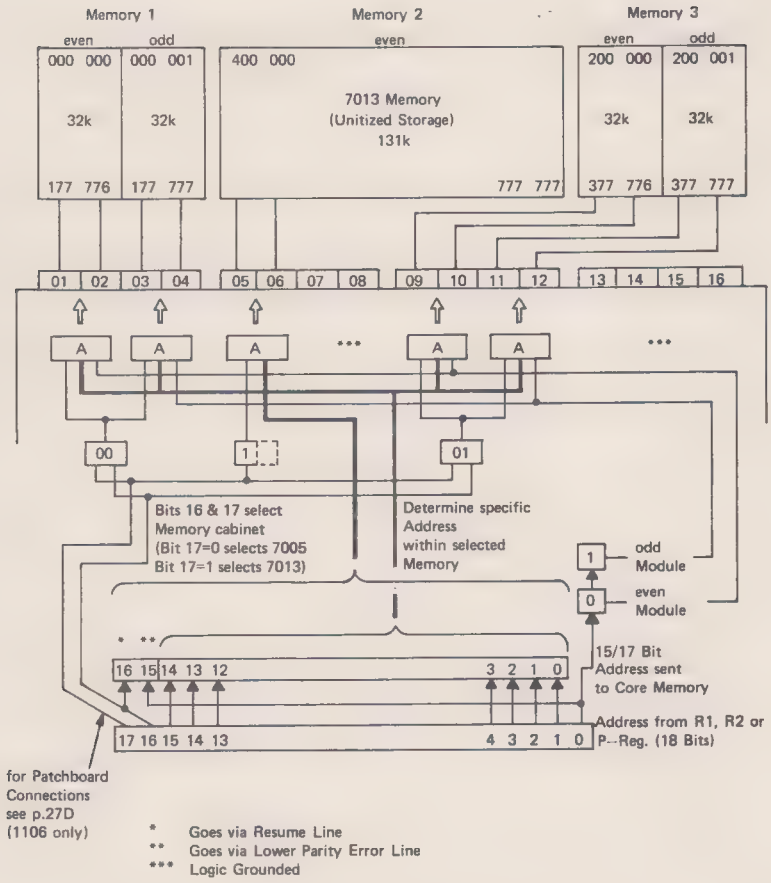
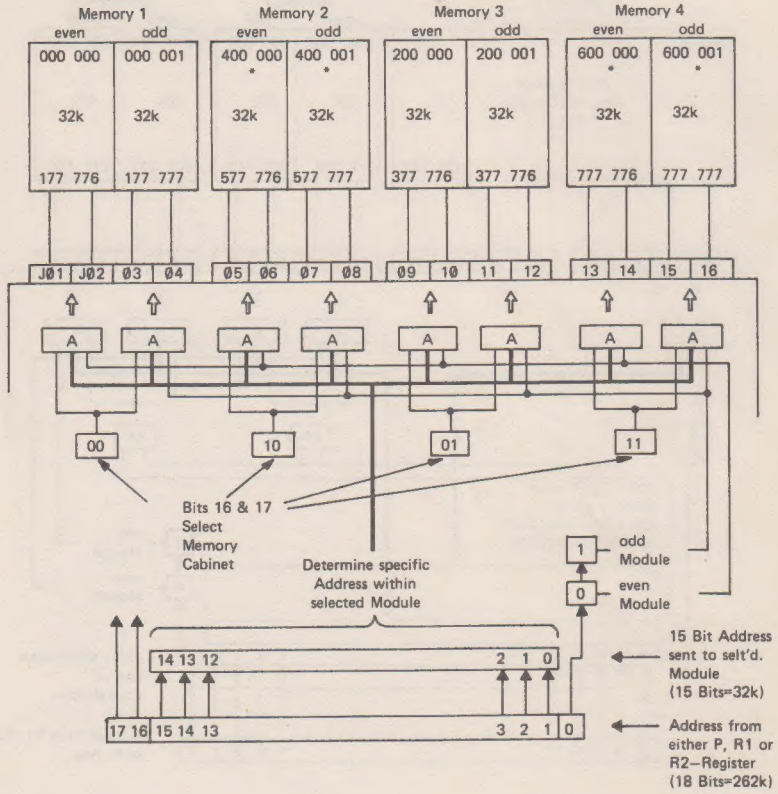
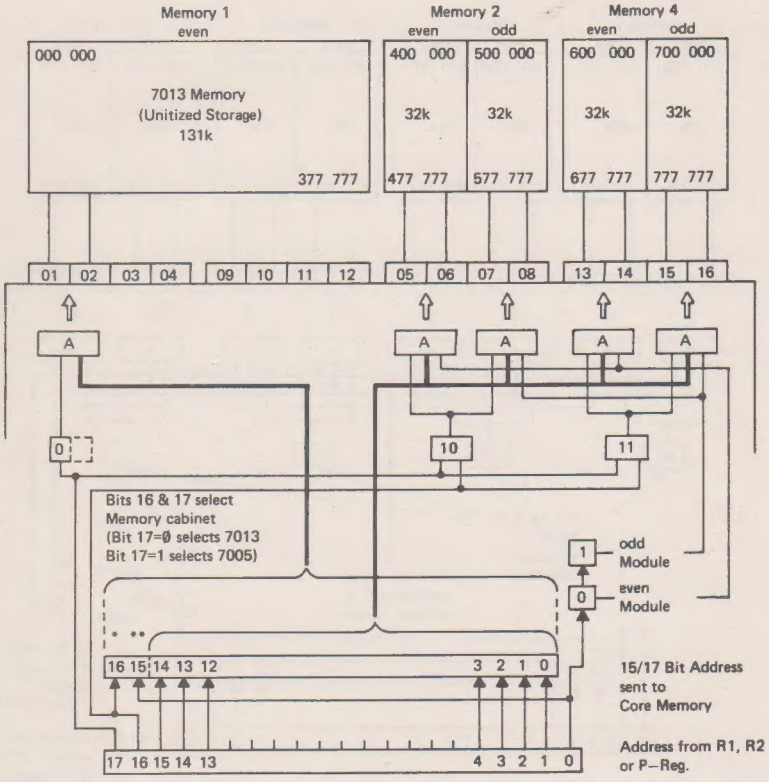


Fig. 8.4: Vertical Mode With Interleaving Feature (7013)



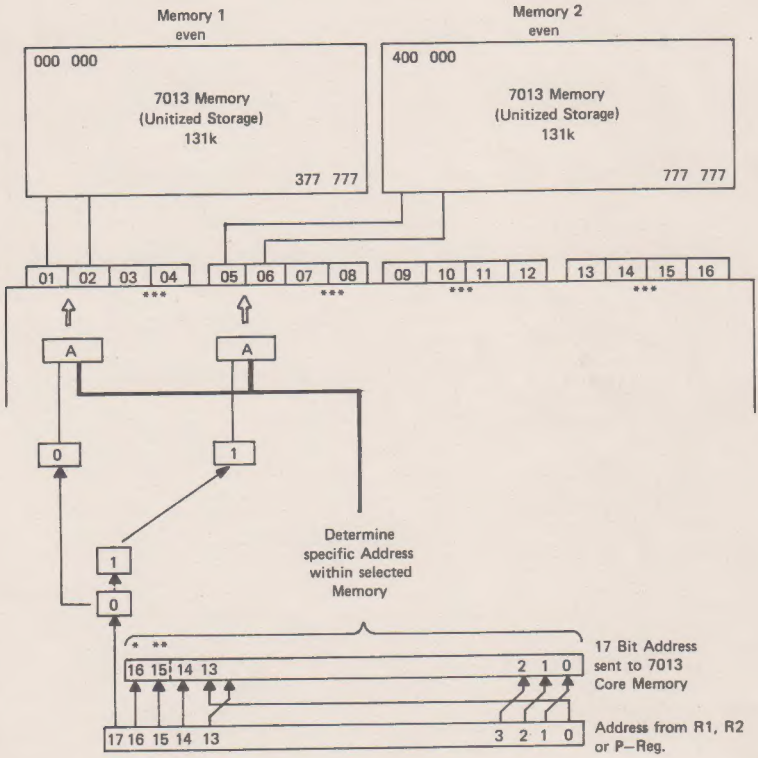
* Maximum Configuration

Fig. 8.5: Unit or Multiprocessor System With Interleaving Feature (F1053–00)



* Goes via Resume Line to 7013
** Goes via Lower Parity Error Line to 7013
*** Logic Grounded

Fig. 8.6: Vertical Mode With Interleaving Feature (7013)



* Goes via 'Resume'–Line
** Goes via 'Lower Parity Error'–Line
*** Logic Grounded

Fig. 8.7: Unit or Vertical Mode Without Interleaving Feature